

Universidad de las Ciencias Informáticas

Facultad 10

Título: “Sistema automatizado para la gestión de las evaluaciones de desempeño de los profesores de la facultad 10 de la Universidad de las Ciencias Informáticas”

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores: Maikel García Sánchez
Yenisel Borges Martínez

Tutor: Ing. Nilet María Soto López

Co-Tutor: Ing. Susel Vázquez Acuña

Ciudad de la Habana. Mayo, 2010

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor:

Yenisel Borges Martínez

Autor:

Maikel García Sánchez

Tutor:

Ing. Nilet María Soto López

Co-Tutor:

Ing. Susel Vázquez Acuña

Resumen

En la Universidad de las Ciencias Informáticas se aplica el sistema para pago adicional, el cual tiene como objetivo elevar los resultados de los trabajadores. Este sistema se centra en la evaluación del desempeño de los trabajadores. En la facultad 10 se lleva a cabo el proceso de evaluación del desempeño de los profesores donde se identificaron deficiencias como complejidad y demora en la gestión de dicho proceso. Es entonces que surge la idea de que la creación de un sistema automatizado que gestione el proceso de evaluación de desempeño de los profesores de la facultad 10 elevará la eficiencia del mismo.

Para materializar la idea anteriormente expresada fue llevada a cabo la presente investigación con el objetivo de elevar la eficiencia en la gestión del proceso de evaluación de desempeño incorporando un sistema automatizado.

Durante la investigación se realizó un estudio del estado del arte referente a los sistemas automatizados vinculados al campo de acción, metodologías de desarrollo de software, herramientas y lenguajes de programación posibles a utilizar para la elaboración del producto. Además se realizó el diseño e implementación del sistema automatizado cuyo nombre es SAGED.

Para la implementación del mismo se determinó utilizar la metodología Programación Extrema, se hizo uso de las herramientas Netbeans, Visual Paradigm y Fireworks 8, del framework symfony, y de los lenguajes de programación PHP, Javascript y html.

El desarrollo de esta investigación tuvo como resultado un producto eficiente que facilita la gestión del proceso de evaluación de desempeño.

PALABRAS CLAVE

Sistema automatizado, evaluación de desempeño, gestión del proceso de evaluación.

Tabla de contenidos

ÍNDICE DE TABLAS	VI
ÍNDICE DE ILUSTRACIONES	X
INTRODUCCIÓN	1
CAPÍTULO 1: ASPECTOS TEÓRICOS QUE FUNDAMENTAN LA INVESTIGACIÓN	5
1.1. Conceptos asociados al dominio del problema	5
1.2. Gestión del proceso de evaluación de desempeño	7
1.2.1 Descripción General	7
1.2.2 Flujo actual del proceso	7
1.2.3 Análisis crítico de la ejecución del proceso	9
1.3. Sistemas automatizados existentes vinculados al campo de acción	9
1.4. Tendencia y tecnología actuales	10
1.4.1 Aplicación Web	10
1.4.2 Lenguajes de Programación	10
1.4.3 Servidores Web	13
1.4.4 Gestor de Base de Datos	14
1.4.5 Framework	17
1.4.6 Metodología de Desarrollo de software	18
1.4.7 Lenguaje Unificado de Modelado	22
1.4.8 Herramientas para el desarrollo del sistema	22
1.5. Justificación de la selección	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	26
2.1 Propuesta de la solución	26
2.2 Personas relacionadas con el sistema	27
2.3 Exploración	29
2.3.1 Historias de usuario	29
2.4 Planificación de la entrega	34
2.4.1 Estimación de esfuerzos por historias de usuario	34
2.4.2 Plan de entregas	35
2.5 Plan de iteraciones	35
2.5.1 Plan de duración de las iteraciones	36
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	38
3.1 Diseño del sistema	38

3.1.1 Tarjetas CRC	38
3.1.2 Diseño de la Base de Datos del Sistema	44
3.2 Fase de implementación	45
3.2.1 Tareas de la programación	45
Iteración 1	47
Iteración 2	52
Iteración 3	56
3.2.2 Diagrama de despliegue	60
CAPÍTULO 4: PRUEBA	62
4.1 Fase de prueba	62
4.1.1 Pruebas de aceptación	62
CONCLUSIONES	83
RECOMENDACIONES	84
REFERENCIA BIBLIOGRÁFICA	85
BIBLIOGRAFÍA	87

Índice de tablas

TABLA 1 PERSONAS RELACIONADAS CON EL SISTEMA.....	28
TABLA 2 AUTENTICAR.....	29
TABLA 3 MOSTRAR DESEMPEÑO.....	30
TABLA 4 ADMINISTRAR TAREAS	30
TABLA 5 ADMINISTRAR DESEMPEÑO	31
TABLA 6 APROBAR EVALUACIÓN	31
TABLA 7 ADMINISTRAR USUARIOS	32
TABLA 8 ADMINISTRAR TIPOS DE TAREAS	32
TABLA 9 ADMINISTRAR SISTEMA	32
TABLA 10 GENERAR REPORTE DE DESEMPEÑO.....	33
TABLA 11 CREAR PLAN DE RESULTADOS	33
TABLA 12 ESTIMACIÓN DE ESFUERZO DE IMPLEMENTACIÓN.....	34
TABLA 13 PLAN DE ENTREGA.....	35
TABLA 14 PLAN DE DURACIÓN DE LAS ITERACIONES	37
TABLA 15 TARJETA CRC USUARIO.....	38
TABLA 16 TARJETA CRC DESEMPEÑO	39
TABLA 17 TARJETA CRC TAREA.....	39
TABLA 18 TARJETA CRC EVALUACIÓN DE TAREA.....	39
TABLA 19 TARJETA CRC AUSENCIA.....	40
TABLA 20 TARJETA CRC INCIDENCIA	40
TABLA 21 TARJETA CRC EVALUACIÓN DE ÁREA	40
TABLA 22 TARJETA CRC EVALUACIÓN DE DESEMPEÑO	41
TABLA 23 TARJETA CRC APROBACIÓN DE EVALUACIÓN	41
TABLA 24 TARJETA CRC CARGO.....	41
TABLA 25 TARJETA CRC PROFESOR.....	42
TABLA 26 TARJETA CRC DEPARTAMENTO	42
TABLA 27 TARJETA CRC ÁREA	42
TABLA 28 TARJETA CRC TIPO DE TAREA.....	43
TABLA 29 TARJETA CRC ACTIVIDADES	43
TABLA 30 TARJETA CRC ROL	43
TABLA 31 TARJETA CRC PÁGINA PRINCIPAL	43
TABLA 32 TARJETA CRC REPORTE.....	44

TABLA 33 TARJETA CRC PLAN DE RESULTADOS	44
TABLA 34 TARJETA CRC ADMINISTRAR PLAN DE RESULTADOS.....	44
TABLA 35 TAREAS POR HISTORIAS DE USUARIOS.....	46
TABLA 36 TAREA 1 DE LA HU “AUTENTICAR”.....	47
TABLA 37 TAREA 2 DE LA HU “AUTENTICAR”.....	48
TABLA 38 TAREA 1 DE LA HU “MOSTRAR DESEMPEÑO”.....	48
TABLA 39 TAREA 2 DE LA HU “MOSTRAR DESEMPEÑO”.....	48
TABLA 40 TAREA 3 DE LA HU “MOSTRAR DESEMPEÑO”.....	49
TABLA 41 TAREA 4 DE LA HU “MOSTRAR DESEMPEÑO”.....	49
TABLA 42 TAREA 1 DE LA HU “ADMINISTRAR TAREAS”.....	50
TABLA 43 TAREA 2 DE LA HU “ADMINISTRAR TAREAS”.....	50
TABLA 44 TAREA 3 DE LA HU “ADMINISTRAR TAREAS”.....	51
TABLA 45 TAREA 4 DE LA HU “ADMINISTRAR TAREAS”.....	51
TABLA 46 TAREA 1 DE LA HU “ADMINISTRAR DESEMPEÑO”.....	52
TABLA 47 TAREA 2 DE LA HU “ADMINISTRAR DESEMPEÑO”.....	52
TABLA 48 TAREA 3 DE LA HU “ADMINISTRAR DESEMPEÑO”.....	53
TABLA 49 TAREA 4 DE LA HU “ADMINISTRAR DESEMPEÑO”.....	53
TABLA 50 TAREA 5 DE LA HU “ADMINISTRAR DESEMPEÑO”.....	54
TABLA 51 TAREA 1 DE LA HU “APROBAR EVALUACIÓN”.....	54
TABLA 52 TAREA 2 DE LA HU “APROBAR EVALUACIÓN”.....	55
TABLA 53 TAREA 1 DE LA HU “ADMINISTRAR USUARIO”.....	55
TABLA 54 TAREA 2 DE LA HU “ADMINISTRAR USUARIO”.....	55
TABLA 55 TAREA 3 DE LA HU “ADMINISTRAR USUARIO”.....	56
TABLA 56 TAREA 1 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”.....	56
TABLA 57 TAREA 2 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”.....	57
TABLA 58 TAREA 3 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”.....	57
TABLA 59 TAREA 4 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”.....	58
TABLA 60 TAREA 1 DE LA HU “ADMINISTRAR SISTEMA”.....	58
TABLA 61 TAREA 2 DE HU “ADMINISTRAR SISTEMA”.....	58
TABLA 62 TAREA 1 DE HU “GENERAR REPORTE”.....	59
TABLA 63 TAREA 2 DE HU “GENERAR REPORTE”.....	59
TABLA 64 TAREA 1 DE HU “CREAR PLAN DE RESULTADOS”.....	60
TABLA 65 TAREA 2 DE HU “CREAR PLAN DE RESULTADOS”.....	60
TABLA 66 PRUEBA 1 DE LA HU “AUTENTICAR”.....	62

TABLA 67 PRUEBA 2 DE LA HU “AUTENTICAR”	63
TABLA 68 PRUEBA 1 DE LA HU “MOSTRAR DESEMPEÑO”	63
TABLA 69 PRUEBA 2 DE LA HU “MOSTRAR DESEMPEÑO”	64
TABLA 70 PRUEBA 3 DE LA HU “MOSTRAR DESEMPEÑO”	64
TABLA 71 PRUEBA 4 DE LA HU “MOSTRAR DESEMPEÑO”	65
TABLA 72 PRUEBA 1 DE LA HU “ADMINISTRAR TAREAS”	65
TABLA 73 PRUEBA 2 DE LA HU “ADMINISTRAR TAREAS”	65
TABLA 74 PRUEBA 3 DE LA HU “ADMINISTRAR TAREAS”	66
TABLA 75 PRUEBA 4 DE LA HU “ADMINISTRAR TAREAS”	66
TABLA 76 PRUEBA 5 DE LA HU “ADMINISTRAR TAREAS”	67
TABLA 77 PRUEBA 6 DE LA HU “ADMINISTRAR TAREAS”	67
TABLA 78 PRUEBA 1 DE LA HU “ADMINISTRAR DESEMPEÑO”	67
TABLA 79 PRUEBA 2 DE LA HU “ADMINISTRAR DESEMPEÑO”	68
TABLA 80 PRUEBA 3 DE LA HU “ADMINISTRAR DESEMPEÑO”	68
TABLA 81 PRUEBA 4 DE LA HU “ADMINISTRAR DESEMPEÑO”	69
TABLA 82 PRUEBA 5 DE LA HU “ADMINISTRAR DESEMPEÑO”	69
TABLA 83 PRUEBA 1 DE LA HU “APROBAR EVALUACIÓN”	70
TABLA 84 PRUEBA 1 DE LA HU “ADMINISTRAR USUARIO”	70
TABLA 85 PRUEBA 2 DE LA HU “ADMINISTRAR USUARIO”	71
TABLA 86 PRUEBA 3 DE LA HU “ADMINISTRAR USUARIO”	71
TABLA 87 PRUEBA 4 DE LA HU “ADMINISTRAR USUARIO”	71
TABLA 88 PRUEBA 5 DE LA HU “ADMINISTRAR USUARIO”	72
TABLA 89 PRUEBA 6 DE LA HU “ADMINISTRAR USUARIO”	72
TABLA 90 PRUEBA 7 DE LA HU “ADMINISTRAR USUARIO”	73
TABLA 91 PRUEBA 8 DE LA HU “ADMINISTRAR USUARIO”	73
TABLA 92 PRUEBA 9 DE LA HU “ADMINISTRAR USUARIO”	73
TABLA 93 PRUEBA 10 DE LA HU “ADMINISTRAR USUARIO”	74
TABLA 94 PRUEBA 1 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	74
TABLA 95 PRUEBA 2 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	75
TABLA 96 PRUEBA 3 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	75
TABLA 97 PRUEBA 4 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	75
TABLA 98 PRUEBA 5 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	76
TABLA 99 PRUEBA 6 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	76
TABLA 100 PRUEBA 7 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	77

TABLA 101 PRUEBA 8 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	77
TABLA 102 PRUEBA 9 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	77
TABLA 103 PRUEBA 10 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	78
TABLA 104 PRUEBA 11 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	78
TABLA 105 PRUEBA 12 DE LA HU “ADMINISTRAR TIPOS DE TAREAS”	79
TABLA 106 PRUEBA 1 DE LA HU “ADMINISTRAR SISTEMA”	79
TABLA 107 PRUEBA 2 DE LA HU “ADMINISTRAR SISTEMA”	79
TABLA 108 PRUEBA 3 DE LA HU “ADMINISTRAR SISTEMA”	80
TABLA 109 PRUEBA 3 DE LA HU “ADMINISTRAR SISTEMA”	80
TABLA 110 PRUEBA 2 DE LA HU “ADMINISTRAR SISTEMA”	81
TABLA 111 PRUEBA 3 DE LA HU “ADMINISTRAR SISTEMA”	81
TABLA 112 PRUEBA 1 DE LA HU “GENERAR REPORTE”	81
TABLA 113 PRUEBA 1 DE LA HU “CREAR PLAN DE RESULTADOS”	82

Índice de ilustraciones

ILUSTRACIÓN 1 MODELO DE DATOS.....	45
ILUSTRACIÓN 2 DIAGRAMA DE DESPLIEGUE.....	61

Introducción

En la actualidad es una realidad que las organizaciones de éxito traten a su personal como una fuente fundamental de competitividad. La satisfacción laboral implica un conjunto de actitudes positivas y motivaciones de los trabajadores, que influirán de una manera significativa en los comportamientos, y desde luego en la obtención de buenos resultados; de ahí la necesidad de conseguir y contar en todo momento con el personal más calificado, motivado y competitivo posible. En ello es esencial la estimulación, considerada como una inversión de la empresa para obtener mejores resultados futuros.

En la Universidad de las Ciencias Informáticas se elevan los ingresos personales de los trabajadores a través del sistema para pago adicional en moneda nacional. Este tiene como objetivo elevar los resultados individuales y colectivos, los indicadores de eficiencia y disciplina en todas las áreas. También tiene como objetivo reconocer la importancia económica y social de la labor que desarrollan los trabajadores.

La evaluación del desempeño laboral de los trabajadores ya sean docentes, investigadores, especialistas, técnicos y otros, constituye el centro del sistema para pago adicional. Cada uno de ellos es evaluado por su jefe administrativo inmediato superior al que está directamente subordinado, incluidos los jefes de grupos, jefes de proyectos, dirigentes y cuadros.

En la facultad 10 se lleva a cabo la elaboración de las evaluaciones de desempeño de los profesores, donde se identifica como principal deficiencia la ausencia de un procedimiento estándar para realizarlo. En este proceso cualitativo cada jefe administrativo tiene un modo de aplicación diferente. Es difícil tener toda la información de cada uno de los profesores de forma organizada sin que se obvie algo importante, ya que el proceso es complejo, pues evalúa el trabajo de 3 meses.

La evaluación de desempeño final se conforma teniendo en cuenta el criterio de varias personas que inciden en la misma. Al no existir una vía rápida para la gestión de dichos criterios, esto provoca que el proceso sea lento.

Debido a las deficiencias que puede tener este proceso de pago existe la posibilidad de una incorrecta evaluación, lo cual traería como consecuencia la insatisfacción del trabajador o el uso indebido de recursos monetarios por una mala aplicación de dicho proceso.

Este trabajo surge como necesidad de dar solución a las situaciones antes expuestas y su principal tarea está encaminada a resolver el siguiente **problema**: ¿Cómo elevar la eficiencia en la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10 de la Universidad de las Ciencias Informáticas?

El **objeto de estudio** de este trabajo es la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10.

Se propone como **objetivo general** elevar la eficiencia en la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10 incorporando un sistema automatizado.

Para dar cumplimiento al objetivo general se trazan los siguientes **objetivos específicos**:

1. Realizar un estudio del reglamento sobre la aplicación del sistema para pago adicional.
2. Caracterizar las herramientas y lenguajes de programación adecuados para el desarrollo de la aplicación.
3. Implementar un proceso modelo para la gestión de las evaluaciones de desempeño de los profesores de la facultad 10.
4. Desarrollar un sistema automatizado que gestione el proceso de evaluación de desempeño.

El **campo de acción** en el cual estará enmarcada la investigación es el procesamiento de datos en la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10.

La **idea a defender** es que la creación de un sistema automatizado que gestione el proceso de evaluación de desempeño de los profesores de la facultad 10 elevará la eficiencia del mismo.

Para llevar a cabo la investigación se plantean las siguientes tareas:

1. Realizar un estudio preliminar de los reglamentos y de los procedimientos actuales de evaluación del desempeño en los distintos departamentos de la facultad.
2. Realizar una indagación en las facultades de la Universidad de las Ciencias Informáticas con el objetivo de identificar si existe algún mecanismo semi o automatizado cuyo objetivo sea el mismo.
3. Caracterizar los aspectos teóricos conceptuales relacionados con los sistemas de gestión y planificación.
4. Definir las metodologías y herramientas posibles a utilizar.
5. Definir un proceso modelo para la evaluación de desempeño de los profesores de la facultad 10.
6. Analizar y diseñar la aplicación.
7. Implementar la aplicación.
8. Realizar pruebas de aceptación a la aplicación.

Para cumplir con las tareas investigativas del trabajo se usan los siguientes **métodos científicos** clasificados en teóricos y empíricos:

Métodos teóricos.

- **Modelación:** Es un método que nos permite la creación de modelos, que no son más que una reproducción simplificada de la realidad y que nos proporcionó una mejor comprensión de las características del proceso de evaluación del desempeño.
- **Analítico Sintético:** Fue usado con el objetivo de llegar a conocer, mediante el análisis de las diferentes teorías y documentos encontrados a lo largo de la presente investigación, la esencia de los fenómenos relacionados con el proceso de gestión de la evaluación de desempeño.

Métodos empíricos.

- **Entrevista:** Empleada con el objetivo de obtener información importante y adquirir una mayor familiarización con respecto a las peculiaridades y características del proceso de gestión de las evaluaciones de desempeño de los profesores en la facultad 10.
- **Encuesta:** Con la aplicación de este método se pudo medir la aceptación del sistema.

A continuación se muestra una breve descripción de la estructura del trabajo de diploma:

Capítulo 1: Aspectos teóricos que fundamentan la investigación.

En este capítulo se trata los aspectos teóricos que contienen los fundamentos necesarios para una eficiente comprensión de los temas tratados en el resto del documento. Además, sirve como soporte teórico a la construcción del sistema. Se brinda una visión general de algunos conceptos relacionados con la gestión del proceso de evaluación de desempeño, y se caracterizan las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

Capítulo 2: Características del sistema

Se presenta una propuesta del sistema a desarrollar, se muestran los requerimientos de la propuesta de solución mediante las historias de usuario que proporcionaran los detalles sobre la estimación del riesgo y una planificación del tiempo que conllevará la implementación de dicha historia de usuario.

Capítulo 3: Diseño e implementación del sistema

En el presente capítulo se muestra el diseño del sistema y se describen los artefactos relacionados con la implementación. Se definen las tarjetas CRC (Contenido, Responsabilidad, Colaboración) como herramienta de reflexión en el diseño de software orientado a objetos y se detallan las iteraciones para el desarrollo del sistema, exponiéndose las tareas generadas en cada una de ellas por cada historia de usuario

Capítulo 4: Pruebas

Se describen las pruebas realizadas a la aplicación. Para ello se diseñan los casos de pruebas de aceptación a los que será sometida la aplicación en cada una de las iteraciones.

Capítulo 1: Aspectos teóricos que fundamentan la investigación

El objetivo fundamental de este capítulo es comentar los aspectos más importantes que van a servir como soporte teórico a la construcción del sistema. En él se brinda una visión general de algunos conceptos relacionados con la gestión del proceso de evaluación de desempeño, necesarios para comprender el tema que se aborda en este trabajo de investigación. Como elemento indispensable se aborda el estudio del arte realizado sobre los sistemas similares existentes, así como las tecnologías en las que se apoyará el desarrollo del sistema, en función de un análisis de las tendencias actuales. También se analiza la posible metodología a utilizar para el desarrollo del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y en algunos casos son necesarias comparaciones que fundamentarán la propuesta final.

Y finalmente se exponen las propuestas de herramientas y tecnologías a utilizar para el desarrollo de la aplicación, teniendo en cuenta las necesidades existentes y el entorno donde se enmarca el objeto de estudio.

1.1. Conceptos asociados al dominio del problema

Con el objetivo de lograr una mayor comprensión de los temas a tratar a continuación se describen un grupo de conceptos asociados al dominio del problema.

En términos generales, por el término de **gestión** se referirá a la acción y al efecto de administrar o gestionar. A través de una gestión se llevarán a cabo diversas diligencias, trámites, las cuales, conducirán al logro de un objetivo determinado. (1)

La gestión es la capacidad de la institución u organización para definir, alcanzar y evaluar sus propósitos administrando los recursos disponibles.

Un **proceso** es un conjunto de las fases sucesivas de un fenómeno natural o de una operación artificial. (2) Un proceso es un conjunto de actividades o eventos que se realizan con un objetivo determinado.

El término **eficiencia** se refiere a la capacidad de disponer de alguien o de algo para conseguir un efecto determinado. (3) La eficiencia es el uso racional de los medios con que se cuenta para alcanzar un determinado objetivo. Se trata de la capacidad de alcanzar los objetivos y metas planificadas en el menor tiempo posible y con el mínimo de recursos, logrando de esta forma su optimización.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

La **evaluación** surge como una herramienta de control mediante el cual se puede conocer el rendimiento de una persona, grupo u organización. Por otro lado el desempeño no es más que cumplir las obligaciones inherentes a una profesión, cargo u oficio. (4)

La **evaluación del desempeño** constituye el control de las obligaciones y tareas cumplidas en un período de tiempo por una persona, grupo u organización con el objetivo de conocer su rendimiento, especialmente los logros y las deficiencias.

El término **automatización** se refiere a una amplia variedad de sistemas y procesos que operan con mínima o sin intervención del ser humano. (5) En otras palabras los sistemas automatizados constituyen sistemas donde se transfieren las tareas realizadas habitualmente por las personas a una conjunto de elementos tecnológicos.

En sentido general "La **información** es un conjunto de datos acerca de algún suceso, hecho o fenómeno, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo". (6)

La **gestión de la información**, es un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración y conservación con el objetivo de obtener la información adecuada en el tiempo oportuno y con los resultados deseados.

Los **sistemas de gestión de la información** surgen entonces con el objetivo de controlar los grandes volúmenes de información de una forma fácil, sencilla, confiable y además para obtener el mayor rendimiento de este recurso. La información es un agente importante en la modificación de las conductas existentes en una organización, su correcta gestión es una herramienta fundamental para la toma de decisiones, la formación del personal, la determinación de los errores y el control de los procesos. (7)

El proceso de evaluación de desempeño tiene su base en el **plan de resultados**, el cual es una herramienta que permite ordenar y sistematizar toda la información referente al mismo. Este es creado individualmente, y está conformado por un conjunto de tareas que brindan un resultado de valor observable en alguna de las áreas claves del desarrollo de la universidad como son: la Formación, la Producción, la Extensión, etc.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

1.2. Gestión del proceso de evaluación de desempeño

1.2.1 Descripción General

En el mes de enero del año 2000 Cuba decide dar los primeros pasos en la informatización de la sociedad, para ello fue creado el Ministerio de la Informática y las Comunicaciones (MIC), el cual tiene la misión fundamental de promover el uso masivo de las tecnologías de la información y las comunicaciones en la economía nacional, la sociedad y ponerla a disposición de los ciudadanos.

Con el objetivo de informatizar todos los sectores de la nación cubana y desarrollar la industria del Software para contribuir al desarrollo económico del país en agosto del 2002 se crea la Universidad de las Ciencias Informáticas (UCI) adscripta al MIC.

En la UCI se elevan los ingresos personales de los trabajadores a través del sistema para pago adicional en moneda nacional. Este sistema tiene como objetivo elevar los resultados los trabajadores, además de reconocer la importancia económica social de la labor que desarrollan. La evaluación del desempeño laboral de los trabajadores constituye el centro del sistema para pago adicional.

La UCI está compuesta por 13 facultades donde cada una cuenta con un decano el cual es el máximo jefe administrativo en dicha área. En las facultades se encuentran distribuidos los profesores por departamentos. Cada departamento tiene un jefe (máximo responsable en esta área), encargado de supervisar y controlar todos los datos, documentos, eventos, cursos y otras actividades, referente a cada uno de los profesores que pertenecen a su departamento y manejan dicha información con el objetivo de obtener resultados y evaluarlos trimestralmente, según su desempeño en Deficiente, Adecuado o Superior.

1.2.2 Flujo actual del proceso

El proceso de evaluación tiene su base en el plan de resultados anual del profesor que es creado individualmente, y está conformado por un grupo de tareas que tributan a un resultado en alguna de las áreas claves del desarrollo de la universidad, por ejemplo la Formación, la Producción, la Extensión, etc. Controlar el cumplimiento de este plan es tarea fundamental del jefe de departamento, el cual en la actualidad lleva todos estos registros haciendo uso de herramientas ofimáticas, específicamente en documentos Word y Excel. Estos documentos constituyen la constancia del grado de cumplimiento, y de la calidad con que se cumplen las tareas asignadas a los profesores; y la asistencia a las actividades que

Capítulo 1. Aspectos teóricos que fundamentan la investigación

se planifiquen por los distintos niveles (Universitario, Facultad, y Departamento); el grado de cumplimiento y la calidad con que el profesor cumpla con su plan de trabajo y tareas asignadas, determinan la evaluación de desempeño.

Para un profesor, la evaluación del desempeño se realiza cada 3 meses, a fin de que puedan observarse resultados concretos que se hayan obtenido en el período. Los períodos a evaluar son: Enero-febrero-Marzo, Abril-Mayo-Junio, Julio-Agosto-Septiembre y Octubre-Noviembre y Diciembre.

Un profesor se subordina a varias personas desde el punto de vista funcional, por ejemplo como miembro de un colectivo de asignatura, se subordina al Jefe de Asignatura, como miembro de un proyecto se subordina a un Líder de Proyecto, etc; esto implica que el desempeño del mismo sea evaluado por distintas personas en cada uno de los frentes de trabajo en los que actúa. Cada tarea cumplida es informada al jefe administrativo al que pertenece.

Una vez cumplido el trimestre el jefe administrativo debe realizar una recopilación de las tareas realizadas e incumplidas por cada profesor en el período a evaluar y obtener un criterio de las distintas personas a las cuales este se subordina. Haciendo uso de la información obtenida, el jefe de administrativo según el desempeño del mismo le propone una evaluación de Deficiente, Adecuado o Superior. En algunos departamentos el encargado de realizar la propuesta de evaluación de desempeño es el jefe de asignatura y/o el jefe de disciplina, mostrándolo luego al jefe administrativo, quien acepta o rechaza esta propuesta dando sus criterios. Así mismo pueden derivarse recomendaciones al desempeño del profesor, las cuales constituirán criterios adicionales a evaluar en su siguiente trimestre.

Esta propuesta de evaluación es presentada al decano de la facultad, el cual aprueba la evaluación en caso de estar de acuerdo, en caso contrario se analiza y la cambia. A los cuadros de la Facultad les propone la evaluación su decano. **[Anexo 1]**

Posteriormente es discutida cada evaluación con los profesores, analizando las deficiencias y los logros alcanzados en el trimestre, además de las recomendaciones que deberá tener en cuenta para su siguiente período evaluativo. **[Anexo 2]**

Estas evaluaciones sirven para la evaluación anual del profesor calificadas en bien (B), regular (R) o mal (M), que se archiva en el expediente del profesor.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

1.2.3 Análisis crítico de la ejecución del proceso

En el desarrollo del proceso de evaluación de desempeño de los profesores en la facultad 10 se identificó que existen tres factores fundamentales que afectan al desarrollo del proceso, los cuales son: complejidad, demora, y procedimientos diferentes.

La gestión del proceso es compleja ya que es difícil recopilar toda la información de cada uno de los profesores de forma organizada, sin que se deje de tener en cuenta alguna información importante en las evaluaciones, pues el proceso evaluativo evalúa el trabajo de 3 meses.

En la gestión del proceso de evaluación existen demoras, ya que en la evaluación de desempeño inciden varias personas, lo que provoca que el jefe administrativo encargado de evaluar, dependa de la evaluaciones dadas por otras personas para determinar la evaluación final, y al no existir una vía regulada para el desarrollo del proceso trae como consecuencias que el desarrollo del proceso evaluativo sea lento al demorarse o no existir en algunos casos el criterio de los evaluadores.

Existen diferentes formas para llevar a cabo la gestión del proceso de evaluación de desempeño, dicho proceso es cualitativo donde cada jefe administrativo tiene un procedimiento interno diferente para determinar la evaluación de desempeño.

Estas deficiencias que contiene la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10 deben ser eliminadas o mejoradas. Cómo gestionar la información es algo que en la actualidad ha tomado prioridad. Las Tecnologías de la Información y la Comunicación, también conocidas como TIC, son el conjunto de tecnologías desarrolladas para gestionar información mediante la utilización de hardware y software como medio de sistema informático.

El uso correcto y eficiente de las Tecnologías de la Información y la Comunicación junto al uso de sistemas para la gestión de la información mejoran continuamente las políticas, los procedimientos y procesos de una organización. La implementación de un sistema de gestión automatizado eficaz puede ayudar a reducir tiempo, además de lograr mejoras continuas.

1.3. Sistemas automatizados existentes vinculados al campo de acción

Mediante un exhaustivo estudio del arte realizado en las diferentes facultades de la Universidad de las Ciencias Informáticas, con el objetivo de encontrar soluciones

Capítulo 1. Aspectos teóricos que fundamentan la investigación

automatizadas que apoyara la gestión de información relacionado con el proceso de evaluación de desempeño de los profesores, se encontró lo siguiente:

En la facultad 15 se está desarrollando un sistema que calcula la evaluación de desempeño del profesor y emite una propuesta de Deficiente, Adecuado y Superior, llevando los aspectos cualitativos a cuantitativos haciendo uso de una fórmula matemática establecida. Se han realizado varios estudios para determinar esta fórmula matemática la cual se basa principalmente en la calidad del cumplimiento de la tarea y en el peso establecido para cada tarea. La misma se está desarrollando en el Sistema de Gestión de Contenidos Joomla.

1.4. Tendencia y tecnología actuales.

Las herramientas, tecnologías y metodologías son componentes fundamentales para la realización cualquier software, por lo que es importante el uso correcto de dichos componentes con el objetivo de obtener productos con la calidad requerida.

1.4.1 Aplicación Web

Las aplicaciones Web son aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de un navegador. En otras palabras, es una aplicación de software que se codifica en un lenguaje soportado por los navegadores web en los que se muestran las respuestas de las peticiones enviadas al servidor por los usuarios.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software.

1.4.2 Lenguajes de Programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. (8)

A continuación se realiza un estudio de los distintos tipos de lenguaje de programación Web atendiendo a que los mismos se dividen, a tono con la propia arquitectura cliente-servidor, en lenguajes del lado del cliente y lenguajes del lado del servidor.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

Lenguajes del lado del cliente.

Los lenguajes de lado cliente son aquellos que pueden ser directamente interpretados por el navegador web. Estos lenguajes no necesitan de un servidor. Dentro de los lenguajes al lado del cliente se encuentran: JavaScript, HTML, Visual Basic Script y Java.

HTML

Sus siglas vienen de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares los cuales tienen esta forma o <P>.

Cada etiqueta tiene un significado, por ejemplo significa que se escriba en negrita (bold) o <P> significa un párrafo. Así que el HTML no es más que una serie de etiquetas que se utilizan para definir hasta cierto punto la forma o estilo que queremos aplicar a nuestro documento HTML, también es usado para referirse al contenido del tipo de MIME texto/HTML o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores). (9)

JavaScript

JavaScript es un lenguaje de programación interpretado, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. JavaScript sirve principalmente para mejorar la gestión de la interfaz cliente/servidor como por ejemplo, un script JavaScript insertado en un documento HTML permite reconocer y tratar localmente los eventos generados por el usuario.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM sus siglas vienen de Document Object Model (Modelo de Objetos de Documento). (10)

CSS

Las hojas de estilo en cascada o (Cascading Style Sheets) son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML. Este estilo propone las ventajas:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea). (11)

Lenguajes del lado del servidor.

Lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato para que este muestre los resultados. Estos son empleados para el diseño de páginas webs dinámicas donde los más utilizados para el desarrollo son: ASP, JSP, PHP y PERL.

PHP

Sus siglas vienen de Hypertext Pre-processor (Pre-Procesador de Hipertextos) es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica. Generalmente se ejecuta en un servidor web, tomando el código PHP como entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los

Capítulo 1. Aspectos teóricos que fundamentan la investigación

sistemas operativos y plataformas sin costo alguno. Es un lenguaje de alta potencia, fácil de usar e incluye la programación orientada a objetos.

Principales usos del PHP son:

- Programación de páginas Web dinámicas, habitualmente en combinación con motores de bases de datos.
- Programación en consola, al estilo de Perl o Shell scripting.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK lo que permite desarrollar aplicaciones de escritorio en los sistemas operativos en los que está soportado.

Ventajas de PHP:

- Es un lenguaje multiplataforma, libre.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Permite crear los formularios para la Web.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida. (12)

1.4.3 Servidores Web

Un servidor web es un programa que implementa el protocolo HTTP sus siglas vienen de Hypertext Transfer Protocol (Protocolo de transferencia de hipertexto) o el protocolo HTTPS (la versión cifrada y autenticada). El protocolo HTTP pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. (13) Es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte

de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Servidor Web Apache

En la actualidad el servidor Web Apache es el más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma.

El servidor Apache es un software que está estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- Módulos Base: Módulo con las funciones básicas del Apache.
- Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

El resto de funcionalidades del servidor se consigue por medio de módulos adicionales que se pueden cargar. Para añadir un conjunto de utilidades al servidor, simplemente hay que añadirle un módulo, de forma que no es necesario volver a instalar el software.

Ventajas de Apache:

- Modular, multiplataforma, extensible
- Open source (Código abierto).
- Fácil de conseguir ayuda/soporte). (14)

1.4.4 Gestor de Base de Datos

Hoy en día existen muchas empresas y sitios web que necesitan mantener de forma eficiente un gran volumen de datos. Muchos de ellos optan por soluciones comerciales, aunque muchas otras confían en el software libre optando por una solución como PostgreSQL o MySQL.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

Estos Sistemas de Gestión de Bases de Datos (SGBD) pueden definirse como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Se componen de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta y tienen como propósito general manejar de manera clara, sencilla y ordenada un conjunto de información. Por lo que las principales funciones que deben cumplir estos SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad.

De forma general se destacan entre ellos las siguientes ventajas:

- Gran velocidad de procesamiento en muy poco tiempo.
- Integridad referencial al terminar los registros.
- Independencia del tratamiento de información.
- Facilidad de manejo de grandes volúmenes de información.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.

MySQL

Sistema de gestión de bases de datos relacional. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- El servidor soporta mensajes de error en distintas lenguas.
- Gestión de usuarios y contraseña, manteniendo un muy buen nivel de seguridad en los datos. (15)

Ventajas:

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad.

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

La última serie de producción es la 8.4, siendo la última versión disponible en el momento actual la 8.4.2.

Sus características técnicas la hacen una de las bases de datos más potentes y robusta del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las

Capítulo 1. Aspectos teóricos que fundamentan la investigación

características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (16)

Ventajas de PostgreSQL:

- Instalación ilimitada, multiplataforma.
- Mejor soporte que los proveedores comerciales
- Ahorros considerables en costos de operación
- Estabilidad y confiabilidad legendarias
- Extensible
- Diseñado para ambientes de alto volumen
- Herramientas gráficas de diseño y administración de bases de datos. (17)

1.4.5 Framework

Framework es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un framework proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un framework facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

¿Qué ventajas tiene utilizar un framework?

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que "rellenar".
- Facilita la colaboración. Cualquiera que haya tenido que "pelearse" con el código fuente de otro programador (o incluso con el propio, pasado algún tiempo) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos. Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo. (18)

En la actualidad existen muchos framework entre los más usados se encuentran Symfony, Mojavi, CakePHP, Zend Framework, Prado, CodeIgniter, PHPOnTrax, Phrame y Seagull.

Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux, entre otras, como en plataformas Windows. A continuación se muestran algunas de sus características:

- Fácil de instalar y configurar en la mayoría de las plataformas.
- Independiente de un sistema gestor de bases datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de convenir en vez de configurar, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la Web.
- Preparado para aplicaciones empresariales y adaptables a las políticas de arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (18)

1.4.6 Metodología de Desarrollo de software

En un proyecto de desarrollo de software, la metodología define “Quién” debe hacer “Qué”, “Cuándo” y “Cómo” debe hacerlo. Una metodología es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto.

Capítulo 1. Aspectos teóricos que fundamentan la investigación

Uno de los principales problemas en la actualidad en el desarrollo de software es seleccionar la metodología más adecuada que posibilite obtener los resultados óptimos que se desean; o sea, cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos a nivel mundial. Una metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Se analizará seguidamente dos de las más conocidas, sus características.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, se han desarrollado dos corrientes en lo referente a los procesos de desarrollo: los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los primeros intentan conseguir el objetivo común por medio de orden y documentación, los segundos (también llamados métodos ágiles) tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen el proceso. (19)

A continuación se detallarán algunas características de las Metodologías RUP y XP, a razón de que hoy día son de las más utilizadas, en vista de una mayor comprensión de la selección realizada.

El Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado de Desarrollo Software (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

RUP está basado en 5 principios clave que son: adaptar el proceso, balancear prioridades, demostrar valor iterativamente, elevar el nivel de abstracción y enfocarse en la calidad.

El ciclo de vida RUP es una implementación del desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases: Inicio, Elaboración, Construcción y Transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor énfasis en las distintas actividades. Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la

Capítulo 1. Aspectos teóricos que fundamentan la investigación

tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base-line (Línea Base) de la arquitectura. En la fase de Construcción, se construye el producto por medio de una serie de iteraciones. En la fase de Transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios. (20)

Principales características de RUP:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso y el código fuente) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (del inglés Unified Model Language), para describir todo el proceso, basándose en componentes. (21)

Programación Extrema (XP)

A diferencia de RUP como metodología pesada, la Programación Extrema (XP) es una metodología ágil que intenta reducir la complejidad del software por medio de un trabajo orientado al objeto, basado en las relaciones interpersonales y la velocidad de reacción. Intenta minimizar el riesgo de fallo del proceso por medio de la disposición permanente de un representante competente del cliente a disposición del equipo de desarrollo. Este

Capítulo 1. Aspectos teóricos que fundamentan la investigación

representante debe estar en condiciones de contestar rápida y correctamente a cualquier pregunta del equipo de desarrollo de forma que no se retrase la toma de decisiones.

XP define UserStories como base del software a desarrollar. A partir de los UserStories y de la arquitectura perseguida se crea un plan de release entre el equipo de desarrollo y el cliente. Para cada release se discutirán los objetivos de la misma con el representante del cliente y se definirán las iteraciones (de pocas semanas de duración). Cada iteración **[Anexo 3]** tendrá como resultado un programa que se le entrega al cliente para que lo juzgue, si el cliente no está de acuerdo se definen las próximas iteraciones del proyecto y se someten a la aprobación del cliente hasta que el cliente esté completamente de acuerdo y el software cumpla con todos sus requisitos. A diferencia de otros métodos, en XP la codificación pertenece al equipo completo, de forma que el conocimiento de la aplicación lo posea el equipo entero y no unos pocos miembros.

En esta metodología se sigue un diseño evolutivo con la siguiente premisa: Conseguir la funcionalidad deseada de la forma más sencilla posible. Este diseño hace que apenas se le dé importancia al análisis como fase independiente, debido a que se trabaja exclusivamente en función de las necesidades del momento. (22)

Ventajas:

- Al trabajar de dos en dos, el código será de mayor calidad desde el mismo momento de crearlo y tendrá menos fallos.
- Los programadores novatos aprenderán de los expertos al emparejarse con ellos.
- Si una pareja realiza un trozo de código susceptible de ser reutilizado en el proyecto, hay dos programadores que lo saben y que lo reutilizarán cuando puedan (ya que saben cómo funciona), enseñándolo a sus nuevos compañeros. De esta manera el conocimiento del código ya hecho se propaga de forma natural entre todos los programadores del equipo.
- El estilo de programación tiende a unificarse.

Inconvenientes:

- Para un programador experto puede resultar tedioso tener a un novato a su lado permanentemente.
- El programador experto no aprende y su trabajo se ve ralentizado.
- La mejora o cambios en el estilo de programación puede resultar más complejo.

1.4.7 Lenguaje Unificado de Modelado.

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML es ya un estándar de la industria del software, pero no solo de la industria sino, que en general, de cualquier industria que requiera la construcción de modelos como condición previa para el diseño y posteriormente para la construcción de prototipos. (23)

UML ha nacido como un lenguaje, pero es mucho más que un lenguaje de programación. En realidad se ha diseñado y construido un lenguaje que ha nacido con una madurez sólida si se le compara, incluso con los últimos desarrollos de HTML, C#, Java, Ajax, Xml, los lenguajes por excelencia del mundo de la Internet. (24)

1.4.8 Herramientas para el desarrollo del sistema.

Zend Studio.

Zend Studio o Zend Development Environment es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre.

Características:

1. No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
2. Soporte para PHP 4 y PHP 5, con phpDoc integrado.
3. Resaltado de sintaxis, autocompletado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.
4. Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).

Capítulo 1. Aspectos teóricos que fundamentan la investigación

5. Inserción automática de paréntesis y corchetes de cierre.
6. Sangrado automático y otras ayudas de formato de código.
7. Emparejamiento (matching) de paréntesis y corchetes (si se sitúa el cursor sobre un paréntesis (corchete) de apertura (cierre), Zend Studio localiza el correspondiente paréntesis (corchete) de cierre (apertura)).
8. Detección de errores de sintaxis en tiempo real.
9. Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
10. Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox.
11. Soporte para gestión de grandes proyectos de desarrollo.
12. Manual de PHP integrado.
13. Soporte para control de versiones usando CVS o Subversión.
14. Cliente FTP integrado.
15. Soporte para navegación en bases de datos y ejecución de consultas SQL.

Zend Studio fue diseñado para usarse con el lenguaje PHP; sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML. (25)

Herramienta CASE de Desarrollo de Software

Estas herramientas son de gran utilidad pues intervienen en todos los aspectos del ciclo de vida del desarrollo del software, permitiendo a los desarrolladores menos complejidad a la hora de realizar diferentes actividades del proceso de software como son la realización del diseño del proyecto, la documentación o detección de errores.

Rational Rose

Es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática, otra dinámica de los modelos del sistema, una lógica y otra física; que permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (26)

A continuación los tipos de modelos:

1. Desarrollo Iterativo
2. Generador de Código
3. Ingeniería Inversa
4. Trabajo en Grupo

Visual Paradigm

Visual Paradigm es una herramienta multiplataforma para el modelado de aplicaciones, sobre todo en proyectos donde se vaya a aplicar intensivamente los conceptos avanzados de orientación a objeto, ejemplo de ello es el uso del lenguaje Java, para el cual es vital realizar un buen diseño del sistema. (26)

Facilita a los ingenieros de software diseñar, integrar y modelar visualmente los distintos diagramas que se generan a lo largo del desarrollo del software. Para construir sistemas de software a gran escala de manera confiable a través del uso de un enfoque Orientado al Objeto. Presenta un generador de código que soporta más de 10 lenguajes y proporciona la ingeniería inversa. (27)

1.5. Justificación de la selección.

Se ha desarrollado un estudio de algunas de las tecnologías, herramientas y lenguajes candidatos para la implementación de la propuesta, mencionando características, ventajas y desventajas que las mismas poseen, la decisión de utilizarlas fue tomada siguiendo fundamentalmente la política de uso de herramientas con soporte multiplataforma y licencias de utilización libre.

A partir de estas consideraciones se usará PHP como lenguaje de programación del lado del servidor, al ser un lenguaje desarrollado en el ámbito de los sistemas libres, bajo la licencia GNU. PHP es un lenguaje orientado a objetos, simple, elegante y con seguridad en el tratamiento de tipos, y cuenta con las principales funciones para lograr una solución factible al problema en cuestión.

JavaScript, XHTML y CSS fueron los lenguajes de programación seleccionados del lado del cliente, ya que JavaScript nos brinda la posibilidad de una mejor interacción con los usuarios a través del manejo de eventos, XHTML permite estructurar y etiquetar los documentos para su mejor organización y visualización, permitiendo además utilizar objetos tales como imágenes dentro del texto, y CSS facilita crear variados estilos de páginas web, definiendo

Capítulo 1. Aspectos teóricos que fundamentan la investigación

desde una plantilla CSS la visualización general de los sitios, haciendo las aplicaciones más atractivas y amigables para los usuarios.

Se escogió además la herramienta Zend Studio como editor para PHP. La elección se sustenta fundamentalmente en su soporte, posibilidades de depuración y pruebas de PHP. Symfony fue el framework seleccionado porque, entre otras razones, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web; proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja; así como automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación y posee como ventaja fundamental, una alta abstracción en el acceso a datos. Habiendo hecho la selección anterior es conveniente utilizar como Servidor Web, el Apache, por su gran potencialidad.

Como sistema de gestión de base de datos, se seleccionó MySQL, debido a que posee gran velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento, además de ser fácil de instalar y configurar.

Después de haber analizado las características fundamentales de las metodologías de desarrollo RUP y XP se determinó que para un mejor desarrollo del proyecto, era necesario utilizar XP, ya que, es un proyecto de corto plazo, es decir pequeño. Con la comparación de las herramientas de modelado CASE se llegó a la conclusión de que se utilizaría Visual Paradigm, la principal razón de esta selección es que es una herramienta libre, multiplataforma.

En este capítulo se realizó un estudio de la base teórica que sustentará el desarrollo del sistema. Se describió el objeto de estudio, así como la situación problemática que da origen al problema científico de esta investigación. Se ha fundamentado además la selección de la tecnología y herramientas a utilizar en el desarrollo del futuro sistema, profundizando en las características principales de cada una de ellas y estableciendo comparaciones entre posibles candidatos que han permitido realizar la elección con mayor efectividad y precisión.

Capítulo 2: Características del sistema

En el presente capítulo se procede con la elaboración de la propuesta del sistema a implementar, abordando en la misma una descripción de los módulos por los que estará compuesta, los diferentes roles establecido en ella y todo lo referente a ello.

También en el mismo se definirán las historias de usuario como elemento fundamental para conocer los requisitos del sistema y ser implementadas en cada una de las iteraciones previstas, mientras el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, se pone a prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Además de definir toda la planificación de entrega para la implementación de las mismas y el plan de iteraciones con su correspondiente descripción

2.1 Propuesta de la solución

Teniendo en cuenta los requerimientos planteados y como medio de cumplimiento a los objetivos propuestos inicialmente se ha concebido como propuesta de solución al problema científico la implementación de una aplicación Web que garantice la gestión del proceso de evaluación de desempeño de los profesores de la facultad 10 en la UCI.

El sistema proporcionará una interfaz web como forma de presentación al usuario, la cual tendrá una página de bienvenida que ofrecerá información general sobre el sistema y desde la cual los usuarios se podrán autenticar.

Una vez autenticado el profesor, la aplicación le proporcionará de acuerdo a su rol una serie de funcionalidades.

Para ello el sistema contará con los siguientes 8 módulos:

Autenticación de usuario, desempeño, administración de tareas operativas, administración de desempeño, crear plan de resultados, generar reporte del desempeño, aprobación de evaluación y administración del sistema.

El módulo Autenticación de usuario consiste en validar el usuario y la contraseña introducida en el dominio UCI, una vez autenticado el usuario puede acceder a los módulos que tenga acceso según su rol.

El módulo desempeño le permitirá observar el desempeño en 4 espacios fundamentales como son: plan de resultados anual, tareas operativas, ausencias a actividades e incidencias.

Haciendo uso del módulo administración de tareas operativas el usuario podrá crearse tareas para ser cumplidas por él ó asignar a una o varias personas una tarea especificando su evaluador. Mediante el módulo administración de desempeño se podrán añadir a los profesores ausencia a diferentes actividades, y reportar sus incidencias, además de emitir una evaluación a sus tareas, así como de su trabajo en general en cada una de las áreas de resultados claves, y proporcionar la propuesta de evaluación final del desempeño de los mismos, para luego apoyándose en el módulo aprobación de evaluación ser aprobada o modificada la propuesta de evaluación.

Cada usuario podrá generar un documento donde se recoja su desempeño en un determinado trimestre apoyándose en el módulo generar reporte de desempeño. También se podrá introducir en la aplicación el plan de resultado anual del profesor haciendo uso del módulo crear plan de resultados.

De manera general el módulo administración del sistema permitirá la gestión de cargo, profesor, departamento, área, tipo de tarea, ausencias, roles, además de configurar el plan de resultados y la página principal de la aplicación.

El sistema contará con la existencia de 6 roles, los cuales son: Profesor, evaluador de tarea, evaluador de áreas, evaluador de desempeño, aprobador de evaluación y administrador del sistema. De acuerdo con el rol que le corresponda el usuario autenticado poseerá ciertos privilegios y permisos que le permitirán tener acceso a las diferentes funcionalidades de la aplicación.

En resumen, con esta propuesta de sistema se facilitará la centralización, manejo y control de la información del proceso de evaluación de desempeño de los profesores de una forma estándar y organizada.

2.2 Personas relacionadas con el sistema

Como elemento indispensable a tener en cuenta cuando se comienza el desarrollo de un sistema informático es delimitar la audiencia a la cual va dirigido el mismo. Teniendo en cuenta que la misma a su vez puede ser dividida en grupos atendiendo a sus necesidades.

Capítulo 2. Características del sistema

Dentro de la audiencia a interactuar con el sistema a desarrollar se incluyen todas aquellas personas que obtiene un resultado de valor de al menos uno de los procesos que se ejecutarán en el mismo.

Tabla 1 Personas relacionadas con el sistema

Personas relacionadas con el sistema	Justificación
Profesor	Es la persona que tiene permisos para ver la información referente a su desempeño, crear su plan de resultados y tareas para ser realizadas por él, y generar en un documento su desempeño.
Evaluador de tarea	Es la persona que podrá evaluar las tareas que se la hayan asignado como evaluador de las mismas.
Evaluador de área	Es la persona que podrá emitir una evaluación del área de resultados clave en la cual es el encargado de evaluar el resultado o desempeño de los profesores.
Evaluador de desempeño	Es la persona que podrá añadirle una ausencia, incidencia o asignarle una tarea a un profesor determinado, además de proponer a los profesores que pertenezcan a su departamento una propuesta de evaluación final de su desempeño.
Aprobador de evaluación	Es la persona que podrá aprobar o modificar las propuestas de evaluación final que emitieron los evaluadores de desempeño a sus profesores, además de acceder a los reportes del proceso evaluativo.
Administrador	Es la persona facultada para la gestión del sistema en general. Podrá gestionar la administración de usuarios, tareas y del sistema.

2.3 Exploración

En esta primera fase de XP se tiene como principal objetivo disponer de suficientes requisitos para ser implementados durante el tiempo fijado para una entrega en una determinada iteración. La misma permite definir el alcance del proyecto. En esta etapa los clientes definen sus necesidades a través de las historias de usuario, y el equipo de desarrollo se familiariza con las herramientas y tecnologías que se utilizarán en el proyecto.

2.3.1 Historias de usuario

Como elemento indispensable en el desarrollo de software se encuentra la administración de requisitos, a la cual XP da cobertura mediante la confección de las historias de usuario, para luego ser desarrolladas en un proceso iterativo e incremental. Estas expresan las necesidades del sistema y tienen el mismo propósito que los casos de uso.

Las mismas son redactadas desde la perspectiva del cliente aunque los desarrolladores pueden brindar ayuda en su identificación. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica, siendo el contenido que ellas abarcan concreto y sencillo.

Las historias de usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conllevará su implementación. El nivel de detalle de las historias de usuario debe ser el mínimo posible que permita hacerse una ligera idea de cuánto costará implementar el sistema.

Durante este proceso se identifican 10 historias de usuarios, las cuales se detallan a continuación.

Tabla 2 Autenticar

Historia de Usuario	
Numero: 1	Nombre: Autenticar
Usuario: Profesor	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1

Descripción: El sistema debe brindar la posibilidad de introducir el usuario y contraseña, con la finalidad de verificar y otorgar los permisos según el rol que se ocupe dentro de la aplicación.

Observaciones: El usuario y contraseña deben pertenecer al dominio UCI

Tabla 3 Mostrar desempeño

Historia de Usuario	
Numero: 2	Nombre: Mostrar desempeño
Usuario: Profesor	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe brindar la posibilidad al usuario de ver su desempeño dividido en 4 espacios fundamentales, los cuales serían: Plan de resultado anual, tareas operativas, ausencias a actividades e incidencias.	
Observaciones:	

Tabla 4 Administrar tareas

Historia de Usuario	
Numero: 3	Nombre: Administrar tareas
Usuario: Profesor, evaluador de desempeño.	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 1
Descripción: El sistema debe permitir a los profesores crear tareas operativas incorporando los datos necesarios, ya sea para ser asignadas a uno o varios profesores en el caso de ser evaluador de desempeño o para ser cumplidas por el mismo, además de modificarlas, eliminarlas y listarlas.	
Observaciones: El sistema brinda opciones en dependencia del rol.	

Tabla 5 Administrar desempeño

Historia de Usuario	
Numero: 4	Nombre: Administrar desempeño
Usuario: Evaluador de desempeño, evaluador de tarea, evaluador de área	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 2
<p>Descripción: El sistema debe permitir agregar a un profesor la descripción de un incumplimiento o incidencia que haya sido reportada, además de añadir una ausencia de uno o varios profesores a una actividad, así como modificarlas y eliminarlas. De igual forma debe proporcionar la posibilidad de evaluar a un determinado profesor sus tareas, así como el desempeño en las áreas de resultado claves y emitir una propuesta de evaluación final del desempeño en un período de trabajo trimestral, además debe posibilitar modificar una evaluación emitida en caso de haber evaluado incorrectamente, así como cancelarla.</p>	
Observaciones: El sistema brinda opciones en dependencia del rol.	

Tabla 6 Aprobar evaluación

Historia de Usuario	
Numero: 5	Nombre: Aprobar evaluación
Usuario: Aprobador de evaluación	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 2
<p>Descripción: El sistema debe posibilitar al aprobador de evaluación aceptar o rechazar la propuesta emitida por los evaluadores de desempeño, así como modificar dicha propuesta a un determinado profesor.</p>	
Observaciones:	

Tabla 7 Administrar usuarios

Historia de Usuario	
Numero: 6	Nombre: Administrar usuarios
Usuario: Administrador	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 2
Descripción: El sistema debe permitir agregar profesores incorporando los datos necesarios en la aplicación, además de configurar los cargos y departamentos, así como modificarlos o eliminarlos en caso de que sea necesario.	
Observaciones:	

Tabla 8 Administrar tipos de tareas

Historia de Usuario	
Numero: 7	Nombre: Administrar tipos de tareas
Usuario: Administrador	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: El sistema debe posibilitar crear áreas de resultados claves, tipos de tareas operativas y para el plan de resultados, y actividades para el control de las ausencias, además de modificarlos o eliminarlos en caso de que sea necesario.	
Observaciones:	

Tabla 9 Administrar sistema

Historia de Usuario	
Numero: 8	Nombre: Administrar sistema
Usuario: Administrador	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto

Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: El sistema debe posibilitar crear y asignar los roles que interactúan en él, y configurar la página principal de la aplicación, además de modificarlos o eliminarlos en caso de que sea necesario.	
Observaciones:	

Tabla 10 Generar reporte de desempeño

Historia de Usuario	
Numero: 9	Nombre: Generar reporte de desempeño
Usuario: Profesor	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: El sistema debe brindar la posibilidad de generar un documento donde se refleje el desempeño del profesor concerniente al trimestre especificado.	
Observaciones:	

Tabla 11 Crear plan de resultados

Historia de Usuario	
Numero: 10	Nombre: Crear plan de resultados
Usuario: Profesor	
Prioridad en Negocio: Medio	Riesgo en Desarrollo: Alto
Puntos de Estimación: 1	Iteración Asignada: 3
Descripción: El sistema debe permitir crear el plan de resultado anual.	
Observaciones:	

2.4 Planificación de la entrega

En esta etapa el cliente establece la prioridad de cada historia de usuario, y correspondientemente los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. También se toman acuerdos sobre el contenido de cada entrega.

Para realizar una acertada estimación de la aplicación es necesario estimar de forma ideal el tiempo que les tomará la codificación de cada una de las historias de usuario. Las condiciones ideales se dan cuando se escribe el código sin distracciones y con una dedicación de tiempo completo. Debido a las características del equipo de desarrollo y de la UCI en general, el tiempo de estimación ideal de las historia del sistema, puede extenderse un poco más por cargas extra y un margen de fallo debido a actividades imprevistas. A partir de la prioridad de las historias se decide cuales de ellas se implementarán en las primeras iteraciones.

2.4.1 Estimación de esfuerzos por historias de usuario

En la siguiente tabla se muestra un resumen de la estimación de esfuerzo de implementación para cada historia de usuario, basado en los puntos de estimación.

Tabla 12 Estimación de esfuerzo de implementación

Nro.	Historia de Usuario	Punto de Estimación
1	Autenticar	1
2	Mostrar desempeño	1
3	Administrar tareas	1
4	Administrar desempeño	1
5	Aprobar evaluación	1
6	Administrar usuarios	1
7	Administrar tipos de tareas	1
8	Administrar sistema	1

9	Generar reporte de desempeño	1
10	Crear plan de resultados	1

2.4.2 Plan de entregas

Las historias de usuario servirán para crear el plan estimado de entregas. Cada una de ellas previamente evaluada en tiempo de desarrollo ideal será agrupada por el cliente en orden de importancia.

A continuación se expone el plan de entrega referente a la aplicación:

Tabla 13 Plan de entrega

Módulo	Historia de usuario	Fecha de entrega
Autenticación de usuario	Autenticar	02/03/2010
Desempeño	Mostrar desempeño	08/03/2010
Administración de tareas operativas	Administrar tareas	14/03/2010
Administración de desempeño	Administrar desempeño	22/03/2010
Aprobación de evaluación	Aprobar evaluación	31/03/2010
Administración del sistema	Administrar usuarios	06/04/2010
	Administrar tipos de tareas	12/04/2010
	Administrar sistema	18/04/2010
Generar reporte del desempeño	Generar reporte de desempeño	26/04/2010
Crear plan de resultados	Crear plan de resultados	06/05/2010

2.5 Plan de iteraciones

Todo proyecto que siga la metodología X.P se ha de dividir en varias iteraciones antes de ser entregado. Luego de que las historias de usuarios fueron descritas e identificadas, así como

estimado el esfuerzo que cada una de ellas conlleva se procede a definir las iteraciones en las que serán diseñadas, implementadas y probadas las mismas.

Iteración 1

En esta iteración se pretende diseñar, implementar y probar algunas de las historias de usuario que más importancia tienen para el cliente, las cuales son: autenticar, mostrar desempeño y administrar tareas, obteniéndose las funcionalidades descritas en las mismas. Una vez que se terminen de implementar y probar todas las funcionalidades de esta iteración, se procederá a liberar un ejecutable, el cual será mostrado al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Iteración 2

En esta iteración se tiene como objetivo diseñar, implementar y probar las historias de usuarios de mayor prioridad que restan y algunas de prioridad media, las cuales son: administrar desempeño, aprobar evaluación y administrar usuarios.

Una vez culminado el diseño, la implementación y la prueba de todas las funcionalidades de esta iteración, se procederá a liberar un ejecutable, el cual será mostrado al cliente con la finalidad de que fluya la retroalimentación de ideas y solventar posibles errores.

Iteración 3

En esta iteración se tiene como objetivo el diseño, la implementación y las pruebas de las historias de usuarios restantes, las cuales son: administrar tareas, administrar sistema, generar reporte y crear plan de resultados.

Una vez que se termine de implementar y probar estas funcionalidades ya tendremos la versión 1.0 de la aplicación, disponiéndola al servicio de los profesores de la facultad, con el objetivo de evaluar su funcionamiento, rendimiento y recuperación ante posibles errores.

2.5.1 Plan de duración de las iteraciones

Como parte del ciclo de vida de un proyecto usando la Metodología XP se crea el plan de duración de cada una de las iteraciones. Este plan tiene como objetivo mostrar la duración de cada iteración, así como el orden en que serán implementadas las historias de usuario en cada una de estas.

Tabla 14 Plan de duración de las iteraciones

Iteraciones	Orden de las Historias de usuario a implementar	Duración total de las iteraciones
1	Autenticar	3 semanas
	Mostrar desempeño	
	Administrar tareas	
2	Administrar desempeño	3 semanas
	Aprobar evaluación	
	Administrar usuarios	
3	Administrar tipos de tareas	4 semanas
	Administrar sistema	
	Generar reporte de desempeño	
	Crear plan de resultados	

En este capítulo se hizo referencia a todo lo concerniente a la fase de exploración y planificación del sistema. Se realizó la descripción de las historias de usuarios. A partir de estas historias se realizó toda la planificación de entrega del software, la misma se efectuó conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada. Finalmente se cuenta con 10 historias de usuarios que serán desarrolladas en 3 iteraciones.

Capítulo 3: Diseño e implementación del sistema

3.1 Diseño del sistema

En la fase de diseño XP propone la idea “The simplest thing that could possibly work”, “Lo más simple que pueda funcionar”, aplicándose para ello las prácticas especializadas que la metodología propone, las cuales inciden directamente en la realización y elaboración del diseño de un software ya que, cuando se tiene un diseño simple se presta mayor atención a codificar.

En cada una de las iteraciones en la fase dicha anteriormente se van definiendo las tarjetas CRC (Contenido, Responsabilidad, Colaboración), como herramienta de reflexión en el diseño de software orientado a objetos. Además, en cada iteración se detallan las tareas generadas por sus correspondientes historias de usuario.

Por otra parte se realiza el diseño de la base de datos, artefacto de suma importancia, producto de que dará soporte a la futura aplicación.

3.1.1 Tarjetas CRC

Para el diseño de la aplicación se utilizarán las tarjetas CRC. Estas tarjetas permiten al programador centrarse en el desarrollo orientado a objetos. Cada tarjeta identifica una clase, sus propiedades y relaciones con otra tarjeta. Con el propósito de disfrutar de un diseño simple y entendible para todos aquellos profesores que se dispongan a manejar la información correspondiente a su evaluación de desempeño, se elaboró una tarjeta representativa para cada clase. A continuación se detallan las tarjetas CRC de la aplicación, generadas en esta fase en su correspondiente iteración.

Iteración 1:

Tabla 15 Tarjeta CRC Usuario

Usuario	
Funcionalidades	Colaboraciones
Autenticación Cerrar sección	Profesor

Tabla 16 Tarjeta CRC Desempeño

Desempeño	
Funcionalidades	Colaboraciones
Listar plan de resultados	Usuario
Listar tareas operativas	Tareas
Listar ausencias a actividades	Plan de resultados
Listar incidencias	Ausencia
	Incidencia

Tabla 17 Tarjeta CRC Tarea

Tarea	
Funcionalidades	Colaboraciones
Adicionar tarea operativa	
Listar tareas operativas	
Eliminar tarea operativa	
Modificar tarea operativa	
Asignar tarea operativa	
Listar tareas operativas asignadas	
Eliminar tarea operativa asignada	
Modificar tarea operativa asignadas	
	Usuario
	Profesor
	Tipo de tarea
	Área

Iteración 2:

Tabla 18 Tarjeta CRC Evaluación de tarea

Evaluación de tarea	
Funcionalidades	Colaboraciones
Evaluar tarea	Usuario
Listar tareas a evaluar	Profesor
Listar tareas evaluadas	Tareas
Listar profesores con tareas a evaluar	Plan de resultados

Listar profesores con tareas evaluadas	
Modificar evaluación de tarea	

Tabla 19 Tarjeta CRC Ausencia

Ausencia	
Funcionalidades	Colaboraciones
Agregar ausencia a una actividad	Usuario
Listar ausencia a actividades	Profesor
Eliminar ausencia a una actividad	Actividad
Modificar ausencia a una actividad	

Tabla 20 Tarjeta CRC Incidencia

Incidencia	
Funcionalidades	Colaboraciones
Agregar incidencia	
Listar incidencias	Usuario
Eliminar incidencia	Profesor
Modificar incidencia	

Tabla 21 Tarjeta CRC Evaluación de área

Evaluación de área	
Funcionalidades	Colaboraciones
Evaluar área	Usuario
Listar áreas a evaluar	Profesor
Listar áreas evaluadas	Tareas
Listar profesores con áreas a evaluar	Evaluación de tarea
Listar profesores con áreas evaluadas	Plan de resultados
Modificar evaluación de área	

Capítulo 3. Diseño e implementación del sistema

Tabla 22 Tarjeta CRC Evaluación de desempeño

Evaluación de desempeño	
Funcionalidades	Colaboraciones
Evaluar desempeño	Usuario
Listar profesores	Tareas
Listar tareas evaluadas de un profesor	Profesor
Modificar evaluación de desempeño	Evaluación de tarea
	Evaluación de área
	Plan de resultados
	Ausencia
	Incidencia

Tabla 23 Tarjeta CRC aprobación de evaluación

Aprobación de evaluación	
Funcionalidades	Colaboraciones
Aprobar evaluación de desempeño	Usuario
Listar profesores	Profesor
Modificar evaluación de desempeño	Tareas
	Evaluación de tarea
	Evaluación de área
	Plan de resultados
	Ausencia
	Incidencia

Tabla 24 Tarjeta CRC Cargo

Cargo	
Funcionalidades	Colaboraciones
Adicionar cargo	
Listar cargos	
Eliminar cargo	

Editar cargo	
--------------	--

Tabla 25 Tarjeta CRC Profesor

Profesor	
Funcionalidades	Colaboraciones
Adicionar profesor Listar profesores Eliminar profesor Editar profesor Buscar profesor	Cargo Departamento

Tabla 26 Tarjeta CRC Departamento

Departamento	
Funcionalidades	Colaboraciones
Adicionar departamento Listar departamento Eliminar departamento Editar departamento	

Iteración 3

Tabla 27 Tarjeta CRC Área

Área	
Funcionalidades	Colaboraciones
Adicionar área Listar área Eliminar área Editar área	

Tabla 28 Tarjeta CRC Tipo de tarea

Tipo de tarea	
Funcionalidades	Colaboraciones
Adicionar tipo de tarea	
Listar tipos de tarea	
Eliminar tipo de tarea	
Editar tipo de tarea	

Tabla 29 Tarjeta CRC Actividades

Actividades	
Funcionalidades	Colaboraciones
Adicionar actividad	
Listar actividades	
Eliminar actividad	
Editar actividad	

Tabla 30 Tarjeta CRC Rol

Rol	
Funcionalidades	Colaboraciones
Adicionar rol	
Listar roles	
Eliminar rol	Profesor
Editar rol	

Tabla 31 Tarjeta CRC Página principal

Página principal	
Funcionalidades	Colaboraciones
Adicionar página principal	
Listar páginas principales	
Eliminar página principal	

Editar página principal	
-------------------------	--

Tabla 32 Tarjeta CRC Reporte

Reporte	
Funcionalidades	Colaboraciones
Generar reporte	Usuario Tareas Evaluación de tarea Evaluación de área Plan de resultados Ausencia Incidencia

Tabla 33 Tarjeta CRC Plan de resultados

Plan de resultados	
Funcionalidades	Colaboraciones
Crear plan de resultados	

Tabla 34 Tarjeta CRC Administrar Plan de resultados

Administrar Plan de resultados	
Funcionalidades	Colaboraciones
Insertar tareas al plan de resultados Editar tareas del plan de resultados Eliminar tareas del plan de resultados	

3.1.2 Diseño de la Base de Datos del Sistema

En cualquier aplicación en la cual se gestione información la base de datos desempeña un papel fundamental, y sobre todo para las aplicaciones web como es el caso de esta, por lo que se hace necesario que los datos se almacenen de forma coherente y organizada, para evitar que dicha información se pierda. Es por esto que se realiza el diseño de la base de

datos en cuestión, para lograr la persistencia de los datos de los usuarios, así como la información de los recursos que se gestionan por los mismos.

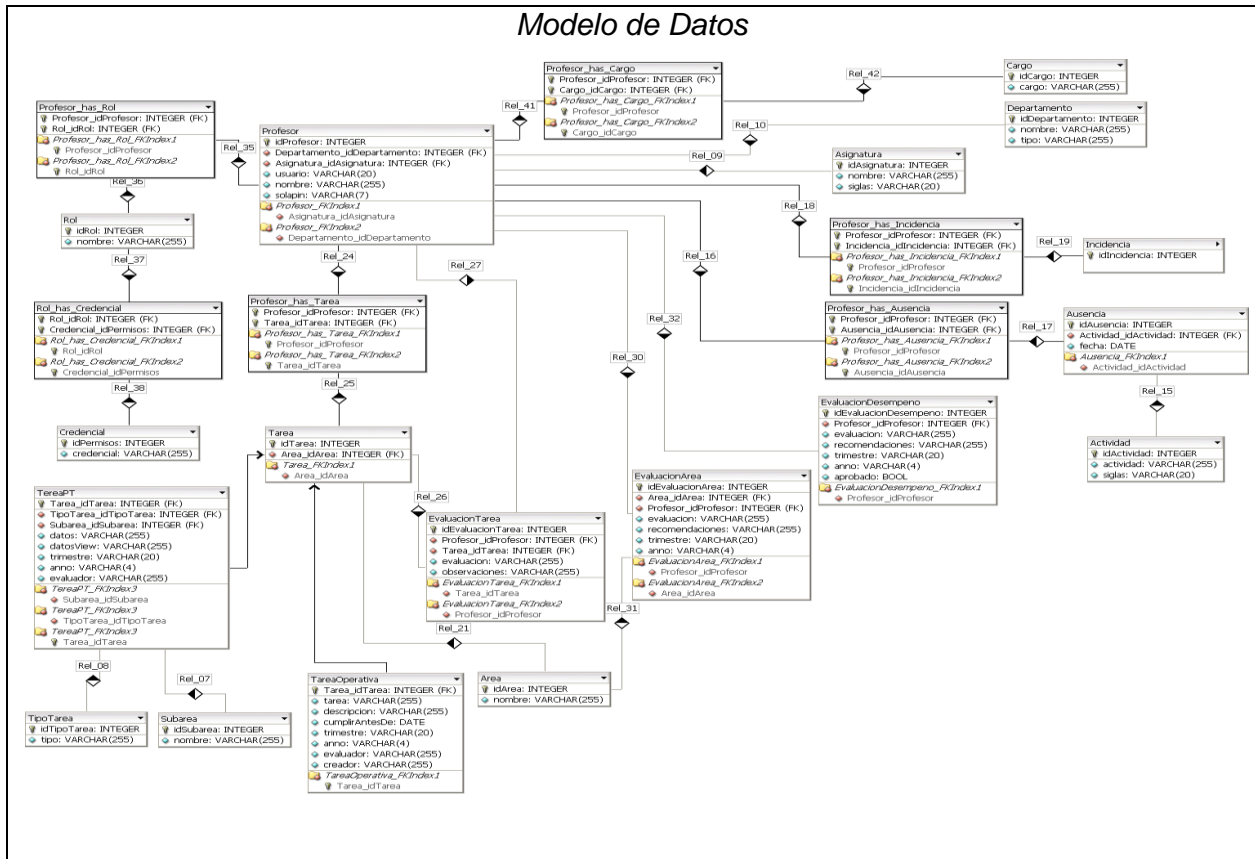


Ilustración 1 Modelo de datos

3.2 Fase de implementación

En esta fase se plantea la implementación de las historias de usuario en su correspondiente iteración, obteniéndose en cada una de ellas una versión funcional del producto. Para ello se descomponen las historias de usuario en tareas de desarrollo para ser realizadas, describiendo cada una de estas.

3.2.1 Tareas de la programación

Cada una de las historias de usuario se transformará en tareas a desarrollar, las cuales serán creadas para ayudar a organizar la implementación exitosa de las mismas. Estas tareas serán para el uso estricto de los programadores. Estas pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente.

Capítulo 3. Diseño e implementación del sistema

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones sobre el sistema, obteniéndose como finalidad un producto funcional con todas las restricciones y características deseadas. A continuación se exponen cada una de las tareas de desarrollo en la que se descompuso cada historia de usuario.

Tabla 35 Tareas por historias de usuarios

Historias de usuario	Tareas
Autenticar	Diseño de la interfaz de autenticación
	Validar datos
Mostrar desempeño	Mostrar tareas cumplidas del Plan de Resultados
	Mostrar tareas operativas cumplidas
	Mostrar control de ausencias
	Mostrar control de incidencias
Administrar tareas	Crear tareas operativas
	Asignar tareas operativas
	Mostrar tareas operativas creadas sin evaluar
	Mostrar tareas operativas asignadas sin evaluar
Administrar desempeño	Añadir ausencias
	Añadir incidencias
	Evaluar tareas
	Evaluar área
	Evaluar desempeño
Aprobar evaluación	Mostrar evaluación
	Aprobar evaluación
Administrar usuario	Administrar cargo

	Administrar profesor
	Administrar departamento
Administrar tipos de tareas	Administrar área
	Administrar tipo de tarea
	Administrar actividades
	Administrar Plan de Resultados
Administrar sistema	Administrar roles
	Administrar página principal
Generar reporte	Generar reporte del Desempeño
	Crear documento del reporte
Crear plan de resultados	Obtener datos
	Mostrar plan de resultados

Iteración 1

En esta iteración se detallan las tareas de algunas de las historias de usuario con alta prioridad en el negocio para el cliente. A continuación se muestran descritas las tareas a desarrollar en la misma.

- **HU “Autenticar”**

Tabla 36 Tarea 1 de la HU “Autenticar”.

Tarea:	
Número de la tarea: 1	Número de HU: 1
Nombre de la tarea: Diseño de la interfaz de autenticación	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 25/02/2010	Fecha fin: 28/02/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	

Descripción: Se implementa el diseño de la interfaz de autenticación, con los componentes necesarios para su funcionamiento.

Tabla 37 Tarea 2 de la HU “Autenticar”.

Tarea:	
Número de la tarea: 2	Número de HU: 1
Nombre de la tarea: Validar datos	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 28/02/2010	Fecha fin: 02/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura de los datos, se verifican que sean correctos con la base de datos de la UCI y la de la aplicación. En caso de que sean correctos se redirecciona al usuario a la página correspondiente según su privilegio.	

- **HU “Mostrar desempeño”**

Tabla 38 Tarea 1 de la HU “Mostrar desempeño”.

Tarea:	
Número de la tarea: 1	Número de HU: 2
Nombre de la tarea: Mostrar tareas evaluadas del Plan de Resultados	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 02/03/2010	Fecha fin: 04/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Mediante una consulta realizada a la base de datos se le muestra al profesor las tareas evaluadas de su plan de resultados.	

Tabla 39 Tarea 2 de la HU “Mostrar desempeño”.

Tarea:	
Número de la tarea: 2	Número de HU: 2

Nombre de la tarea: Mostrar tareas operativas evaluadas	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 04/03/2010	Fecha fin: 06/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se le muestra al profesor las tareas operativas con sus respectivas evaluaciones y los datos necesarios.	

Tabla 40 Tarea 3 de la HU "Mostrar desempeño".

Tarea:	
Número de la tarea: 3	Número de HU: 2
Nombre de la tarea: Mostrar control de ausencias	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 06/03/2010	Fecha fin: 07/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se le muestra al profesor el listado de las actividades a las que ha dejado de asistir hasta ese momento. De cada ausencia se muestran los siguientes datos: actividad a la cual se ausenta, fecha y una observación.	

Tabla 41 Tarea 4 de la HU "Mostrar desempeño".

Tarea:	
Número de la tarea: 4	Número de HU: 2
Nombre de la tarea: Mostrar control de incidencias	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 07/03/2010	Fecha fin: 08/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se le lista al profesor las incidencias que hasta ese momento le han sido reportadas. De cada incidencia se muestran los siguientes datos: incidencia,	

fecha y una observación.

- HU “Administrar tareas”

Tabla 42 Tarea 1 de la HU “Administrar tareas”.

Tarea:	
Número de la tarea: 1	Número de HU: 3
Nombre de la tarea: Crear tareas operativas	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 08/03/2010	Fecha fin: 10/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
<p>Descripción: Se implementan las funcionalidades que permiten la captura de los datos (nombre, cumplir antes de, área, trimestre, evaluador y una breve descripción) de la tarea operativa a crear. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la forma esperada, sin dejarse algún campo del formulario requerido vacío. Se implementan funcionalidades tales como listar las tareas que han sido creadas por el profesor, modificar datos de alguna o eliminarla si aún no ha sido evaluada (si no han sido evaluadas).</p>	

Tabla 43 Tarea 2 de la HU “Administrar tareas”.

Tarea:	
Número de la tarea: 2	Número de HU: 3
Nombre de la tarea: Asignar tareas operativas	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 10/03/2010	Fecha fin: 12/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
<p>Descripción: Se implementan las funcionalidades que permiten la captura de los datos (nombre, cumplir antes de, área, trimestre, evaluador, evaluado(s) y una breve descripción) de la tarea operativa a asignar. Luego se realiza la validación de los</p>	

datos, de forma tal que los mismos sean insertados de la forma esperada, sin dejarse algún campo del formulario requerido vacío. Se implementan funcionalidades tales como listar las tareas que han sido asignadas por dicho profesor, modificar los datos de una estas o eliminarla (si no han sido evaluadas).

Tabla 44 Tarea 3 de la HU “Administrar tareas”.

Tarea:	
Número de la tarea: 3	Número de HU: 3
Nombre de la tarea: Mostrar tareas operativas creadas sin evaluar	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 12/03/2010	Fecha fin: 13/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar las tareas creadas por el profesor que no han sido evaluadas para ser cumplidas por el mismo con los datos (nombre, cumplir antes de, área, trimestre, evaluador y una breve descripción) de dicha tarea.	

Tabla 45 Tarea 4 de la HU “Administrar tareas”.

Tarea:	
Número de la tarea: 4	Número de HU: 3
Nombre de la tarea: Mostrar tareas operativas asignadas sin evaluar	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 13/03/2010	Fecha fin: 14/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar las tareas asignadas por el profesor que no han sido evaluadas con los datos (nombre, cumplir antes de, área, trimestre, evaluador, evaluado(s) y una breve descripción) de la tarea en cuestión.	

Iteración 2

En esta iteración se detallan las tareas las historias de usuario de mayor prioridad que restan y algunas de las de prioridad media. A continuación se muestra descritas las tareas a desarrollar en esta iteración.

- HU “Administrar desempeño”

Tabla 46 Tarea 1 de la HU “Administrar desempeño”.

Tarea:	
Número de la tarea: 1	Número de HU: 4
Nombre de la tarea: Añadir ausencias	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 14/03/2010	Fecha fin: 16/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
<p>Descripción: Se implementan las funcionalidades que permiten la captura de los datos de las ausencias a actividades a reportar, esto serían los siguientes: actividad a la cual se ausentó, fecha, una observación y el (los) profesor (es) que se ausentó a dicha actividad. Luego se realiza la validación de los datos, de forma tal que los mismos sean insertados de la forma correcta, sin dejarse algún campo del formulario requerido vacío. Se implementan funcionalidades tales como listar las ausencias que han sido introducidas por dicho profesor, modificar los datos de una estas o eliminarla.</p>	

Tabla 47 Tarea 2 de la HU “Administrar desempeño”.

Tarea:	
Número de la tarea: 2	Número de HU: 4
Nombre de la tarea: Incidencias	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 16/03/2010	Fecha fin: 17/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
<p>Descripción: Se implementan las funcionalidades que permiten obtener los datos de</p>	

las incidencias a reportar, los cuales serían: incidencia, fecha en que se cometió, una observación de la misma y el (los) profesor (es) que incidieron. A continuación se realiza la validación de los datos insertados, de manera tal que los mismos sean insertados de la forma esperada, sin dejarse algún campo del formulario requerido vacío. Se realizan funcionalidades tales como mostrar la lista de las incidencias que han sido insertadas por dicho profesor, modificar o eliminar alguna de estas.

Tabla 48 Tarea 3 de la HU “Administrar desempeño”.

Tarea:	
Número de la tarea: 3	Número de HU: 4
Nombre de la tarea: Evaluar tareas	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 17/03/2010	Fecha fin: 18/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se desarrollan las funcionalidades que posibilitan la captura de los datos que corresponden a la evaluación de las tareas. Se implementan funcionalidades como listar las evaluaciones que han sido introducidas por ese evaluador de tareas, modificar alguna evaluación añadida por el mismo o eliminarla (si no se ha emitido la evaluación de desempeño).	

Tabla 49 Tarea 4 de la HU “Administrar desempeño”.

Tarea:	
Número de la tarea: 4	Número de HU: 4
Nombre de la tarea: Evaluar áreas	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 19/03/2010	Fecha fin: 20/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que posibilitan mostrar el listado de las tareas evaluadas de cada profesor en el área a evaluar con sus respectivas	

evaluaciones y evaluar dicha área, así como modificar la evaluación (si no se ha emitido la evaluación de desempeño).

Tabla 50 Tarea 5 de la HU “Administrar desempeño”.

Tarea:	
Número de la tarea: 5	Número de HU: 4
Nombre de la tarea: Evaluar desempeño	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 21/03/2010	Fecha fin: 22/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se desarrollan las funcionalidades que permiten mostrar el desempeño del trimestre a evaluar del profesor seleccionado e insertar una propuesta de evaluación final, así como modificar dicha evaluación.	

- HU “Aprobar evaluación”

Tabla 51 Tarea 1 de la HU “Aprobar evaluación”.

Tarea:	
Número de la tarea: 1	Número de HU: 5
Nombre de la tarea: Mostrar evaluación	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 22/03/2010	Fecha fin: 25/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar todos los profesores a los cual se le debe aprobar la evaluación, así como su desempeño y la evaluación propuesta por el evaluador de desempeño.	

Capítulo 3. Diseño e implementación del sistema

Tabla 52 Tarea 2 de la HU “Aprobar evaluación”.

Tarea:	
Número de la tarea: 2	Número de HU: 5
Nombre de la tarea: Aprobar evaluación	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 27/03/2010	Fecha fin: 31/03/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se realizan las funcionalidades que posibilitan aprobar o rechazar la propuesta de evaluación de desempeño del profesor seleccionado, además de modificar dicha evaluación.	

- HU “Administrar usuario”

Tabla 53 Tarea 1 de la HU “Administrar usuario”.

Tarea:	
Número de la tarea: 1	Número de HU: 6
Nombre de la tarea: Administrar cargo	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 31/03/2010	Fecha fin: 02/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura del cargo que se desea insertar. Se implementan: listar cargos, modificar un cargo o eliminarlo.	

Tabla 54 Tarea 2 de la HU “Administrar usuario”.

Tarea:	
Número de la tarea: 2	Número de HU: 6
Nombre de la tarea: Administrar profesor	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 02/04/2010	Fecha fin: 04/04/2010

Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez
Descripción: Se implementan las funcionalidades que permiten la obtención de los datos correspondiente al profesor que se desea insertar, los cuales serían: usuario, nombre, cargo, solapín y departamento. A continuación se realiza la validación de los datos, de manera tal que los mismos sean insertados de la forma correcta, sin que se deje algún campo del formulario requerido vacío. Se desarrollan funciones tales como listar profesores, modificar datos del mismo o eliminarlo.

Tabla 55 Tarea 3 de la HU "Administrar usuario".

Tarea:	
Número de la tarea: 3	Número de HU: 6
Nombre de la tarea: Administrar departamento	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 04/04/2010	Fecha fin: 05/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura de los datos correspondiente al departamento que se desea añadir. Luego se realiza la verificación de los datos, de manera tal que sean insertados de la forma esperada, sin dejarse algún campo del formulario requerido vacío. Se implementan funcionalidades tales como listar departamentos, modificar datos del mismo o eliminarlo.	

Iteración 3

En esta iteración se detallan las tareas las historias de usuario de prioridad media que restan. A continuación se muestra descritas las tareas a desarrollar en la misma.

- HU "Administrar tipos de tareas"

Tabla 56 Tarea 1 de la HU "Administrar tipos de tareas".

Tarea:	
Número de la tarea: 1	Número de HU: 7

Nombre de la tarea: Administrar área	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 06/04/2010	Fecha fin: 08/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura del área que se desea crear. Se implementan funcionalidades tales como listar áreas, modificar un área o eliminarla.	

Tabla 57 Tarea 2 de la HU “Administrar tipos de tareas”.

Tarea:	
Número de la tarea: 2	Número de HU: 7
Nombre de la tarea: Administrar tipo de tarea	
Tipos de tarea: Desarrollo	Puntos estimados: 0.2
Fecha inicio: 08/04/2010	Fecha fin: 10/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura del tipo de tarea que se desea insertar. Se implementan funciones tales como listar tareas, modificar una tarea o eliminarla.	

Tabla 58 Tarea 3 de la HU “Administrar tipos de tareas”.

Tarea:	
Número de la tarea: 3	Número de HU: 7
Nombre de la tarea: Administrar actividades	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 10/04/2010	Fecha fin: 11/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura el tipo de actividad que se desea insertar. Luego de su validación se implementan funciones	

tales como listar actividades, modificarla o eliminarla.

Tabla 59 Tarea 4 de la HU “Administrar tipos de tareas”.

Tarea:	
Número de la tarea: 4	Número de HU: 7
Nombre de la tarea: Administrar Plan de Resultados	
Tipos de tarea: Desarrollo	Puntos estimados: 0.1
Fecha inicio: 11/04/2010	Fecha fin: 12/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura de los tipos de tareas a tener en cuenta en el plan de resultados, así como los datos de las mismas. Luego de su validación se implementan funciones tales como listar las tareas en cuestión, modificar una de ellas o eliminarla.	

- HU “Administrar sistema”

Tabla 60 Tarea 1 de la HU “Administrar sistema”.

Tarea:	
Número de la tarea: 1	Número de HU: 8
Nombre de la tarea: Administrar roles	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 12/04/2010	Fecha fin: 15/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la captura del rol que se desea insertar. Luego se realiza la validación del mismo. Se implementan funciones tales como listar roles, modificar uno o eliminarlo.	

Tabla 61 Tarea 2 de HU “Administrar sistema”.

Tarea:	
Número de la tarea: 2	Número de HU: 8

Nombre de la tarea: Administrar página principal	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 15/04/2010	Fecha fin: 18/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten la capturar el contenido de la página principal que se desea insertar. Se implementan funcionalidades tales modificarla o eliminarla.	

- HU “Generar Reporte”

Tabla 62 Tarea 1 de HU “Generar Reporte”.

Tarea:	
Número de la tarea: 1	Número de HU: 9
Nombre de la tarea: Generar reporte del desempeño	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 19/04/2010	Fecha fin: 22/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten obtener todo el desempeño del profesor en el trimestre seleccionado.	

Tabla 63 Tarea 2 de HU “Generar Reporte”.

Tarea:	
Número de la tarea: 2	Número de HU: 9
Nombre de la tarea: Crear documento del reporte	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 23/04/2010	Fecha fin: 26/04/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten generar un documento en formato .pdf con la información del reporte.	

- HU “Crear plan de resultados”

Tabla 64 Tarea 1 de HU “Crear plan de resultados”.

Tarea:	
Número de la tarea: 1	Número de HU: 10
Nombre de la tarea: Obtener datos	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 30/04/2010	Fecha fin: 02/05/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten obtener los datos referentes al plan de resultado.	

Tabla 65 Tarea 2 de HU “Crear plan de resultados”.

Tarea:	
Número de la tarea: 2	Número de HU: 10
Nombre de la tarea: Mostrar plan de resultados	
Tipos de tarea: Desarrollo	Puntos estimados: 0.3
Fecha inicio: 03/05/2010	Fecha fin: 06/05/2010
Programador responsable: Yenisel Borges Martínez - - Maikel García Sánchez	
Descripción: Se implementan las funcionalidades que permiten mostrar los datos referentes al plan de resultados.	

3.2.2 Diagrama de despliegue

Para modelar el hardware utilizado en las implementaciones del sistema y las relaciones entre sus componentes se utiliza el Diagrama de Despliegue, ya que este permite apreciar de forma visual cómo se encuentran relacionados físicamente los componentes de la aplicación.

En este caso la aplicación se encuentra hospedada en un servidor Web y esta se comunica con un sistema de gestión de base de datos (MySQL).

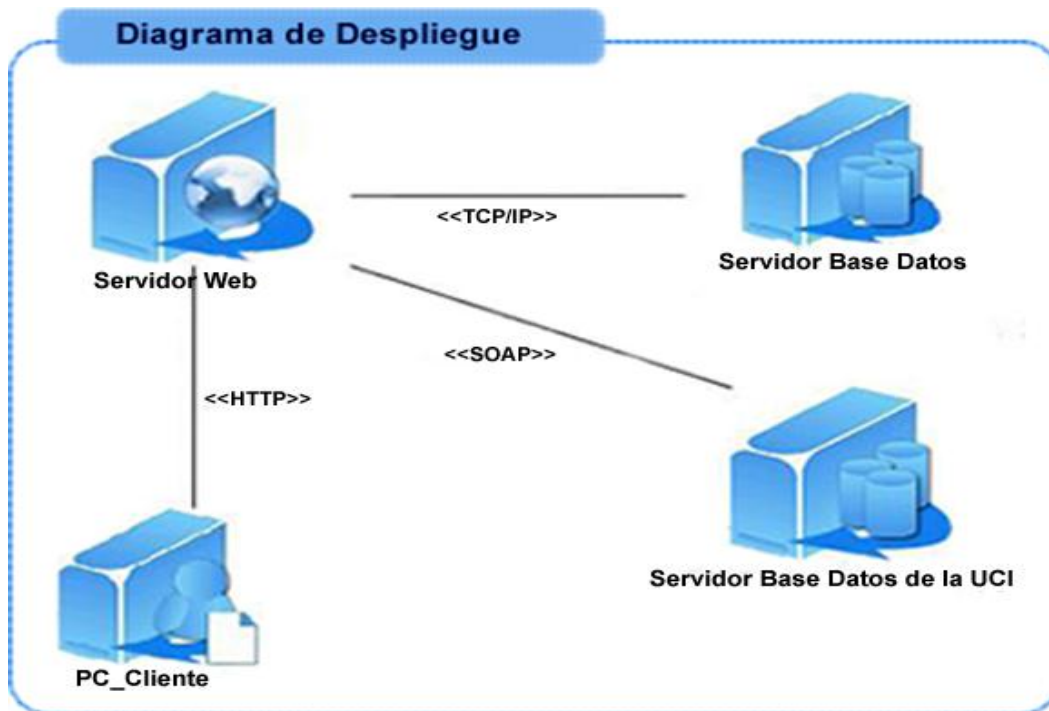


Ilustración 2 Diagrama de despliegue

En resumen, en esta fase se referencia a todo lo concerniente a la fase de Diseño e Implementación, asumiéndose en iteraciones como la metodología utilizada señala. Para ello se realizó la descripción de las tarjetas CRC, logrando un enfoque en la programación orientada a objetos, y de esta forma garantizando una mejor calidad del código en la fase de implementación, y se descompusieron las historias de usuarios en tareas de ingeniería a las cuales se le asignaron tiempos de implementación que fueron cumplidos cabalmente garantizando el objetivo principal de su realización.

Capítulo 4: Prueba

4.1 Fase de prueba

La fase de prueba de la metodología XP anima a frecuentemente estar probando tanto como sea posible. Para ello en cada una de las iteraciones se lleva a cabo los casos de pruebas de aceptación a los que se debe someter la aplicación. Estas pruebas deben ser verificadas en cada entrega que se realice del producto en la planificación establecida.

4.1.1 Pruebas de aceptación

Como técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento son realizadas las pruebas de aceptación. Las cuales son creadas a partir de las historias de usuario y desde la perspectiva del cliente. Una historia de usuario puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento.

Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

El objetivo final de las mismas es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

- **HU “Autenticar”**

Tabla 66 Prueba 1 de la HU “Autenticar”.

Caso de prueba de aceptación:	
Código: HU1_p1	Historia de Usuario: 1
Nombre: Autenticar con datos no válidos	
Descripción: Probar que se pueda autenticar un usuario en el sistema con al menos un dato no válido.	
Condiciones de ejecución: Usuario y/o contraseña no puede(en) ser válido(s)	

Entrada/ Pasos de ejecución: Intentar acceder al sistema con al menos un dato (Usuario y/o contraseña) no válido(Entrar solo uno de ellos y dejar el otro vacío, o que Usuario y/o contraseña no pertenezca al dominio UCI, o pertenezca al dominio UCI y no pertenezca al listado de profesores de la aplicación).
Resultado esperado: El sistema informa al usuario que su usuario y/o contraseña no son válidos, y se le dé la posibilidad de intentarlo nuevamente.
Evaluación de la prueba: Satisfactoria

Tabla 67 Prueba 2 de la HU “Autenticar”.

Caso de prueba de aceptación:	
Código: HU1_p2	Historia de Usuario: 1
Nombre: Autenticar con datos válidos	
Descripción: Probar que se pueda autenticar un usuario en el sistema con datos válidos	
Condiciones de ejecución: Usuario y contraseña deben ser válidos.	
Entrada/ Pasos de ejecución: Intentar acceder al sistema con usuario y contraseña válidos.	
Resultado esperado: Se le muestra una página al usuario según sus privilegios, en la cual se le brinda las funcionalidades correspondientes a su rol.	
Evaluación de la prueba: Satisfactoria	

- HU “Mostrar desempeño”

Tabla 68 Prueba 1 de la HU “Mostrar desempeño”.

Caso de prueba de aceptación:	
Código: HU2_p1	Historia de Usuario: 2
Nombre: Mostrar el plan de resultado anual	
Descripción: Probar que se le muestre al usuario de su desempeño las tareas que han sido cumplidas pertenecientes al trimestre seleccionado.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Intentar ver el plan de resultado.	

Resultado esperado: Se le muestra al usuario todas las tareas que ha cumplido pertenecientes al trimestre seleccionado.
Evaluación de la prueba: Satisfactoria

Tabla 69 Prueba 2 de la HU “Mostrar desempeño”.

Caso de prueba de aceptación:	
Código: HU2_p2	Historia de Usuario: 2
Nombre: Mostrar tareas operativas cumplidas	
Descripción: Probar que se le muestre al usuario el listado de las tareas operativa que han sido cumplidas por él.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Intentar ver las tareas operativas.	
Resultado esperado: Se le muestra al usuario un listado de todas sus tareas operativas cumplidas con sus respectivas evaluaciones.	
Evaluación de la prueba: Satisfactoria	

Tabla 70 Prueba 3 de la HU “Mostrar desempeño”.

Caso de prueba de aceptación:	
Código: HU2_p3	Historia de Usuario: 2
Nombre: Mostrar control de ausencias	
Descripción: Probar que se le muestre al usuario el listado de las actividades a las que se ha ausentado.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Intentar ver las actividades a las que se ha ausentado.	
Resultado esperado: Se le muestra al usuario el listado de las actividades a las que se ha ausentado con sus respectivos datos.	
Evaluación de la prueba: Satisfactoria	

Tabla 71 Prueba 4 de la HU “Mostrar desempeño”.

Caso de prueba de aceptación:	
Código: HU2_p4	Historia de Usuario: 2
Nombre: Mostrar incidencias	
Descripción: Probar que se le muestre al usuario el listado de las incidencias le han sido reportadas.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Intentar ver sus incidencias	
Resultado esperado: Se le muestra al usuario el listado de sus incidencias.	
Evaluación de la prueba: Satisfactoria	

- HU “Administrar tareas”

Tabla 72 Prueba 1 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p1	Historia de Usuario: 3
Nombre: Crear tareas	
Descripción: Probar la funcionalidad crear tarea para ser cumplidas por el mismo.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta crear una tarea insertando los datos correctamente (sin dejar algún campo requerido en blanco).	
Resultado esperado: Se le muestra al usuario la nueva tarea incluida al listado de las tareas que el mismo se ha insertado que no han sido evaluadas.	
Evaluación de la prueba: Satisfactoria	

Tabla 73 Prueba 2 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p2	Historia de Usuario: 3
Nombre: Modificar tarea	
Descripción: Probar la funcionalidad modificar una tarea creada por el usuario.	

Condiciones de ejecución: El usuario debe estar autenticado.
Entrada/ Pasos de ejecución: Se intenta modificar la tarea seleccionada
Resultado esperado: Se muestra la tarea con el cambio realizado en el listado de tareas creadas por el mismo.
Evaluación de la prueba: Satisfactoria

Tabla 74 Prueba 3 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p3	Historia de Usuario: 3
Nombre: Eliminar tarea	
Descripción: Probar la funcionalidad eliminar una tarea creada por el usuario.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta eliminar una tarea seleccionada	
Resultado esperado: Se muestra el listado de tareas creadas por el usuario sin la tarea que se eliminó.	
Evaluación de la prueba: Satisfactoria	

Tabla 75 Prueba 4 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p4	Historia de Usuario: 3
Nombre: Asignar tareas	
Descripción: Probar la funcionalidad asignar tarea a determinado(s) profesor(es).	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de desempeño.	
Entrada/ Pasos de ejecución: Se intenta crear una tarea para asignar a uno o varios profesores, para ello se insertan los datos correctamente (sin dejar algún campo requerido en blanco).	
Resultado esperado: Se le muestra al usuario la tarea que él asignó en el listado de las tareas que ha asignado a profesores y no han sido evaluadas.	
Evaluación de la prueba: Satisfactoria	

Tabla 76 Prueba 5 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p5	Historia de Usuario: 3
Nombre: Modificar una tarea asignada	
Descripción: Probar la funcionalidad modificar una tarea asignada por el usuario.	
Condiciones de ejecución: El usuario debe estar autenticado y tener los privilegios del evaluador de desempeño.	
Entrada/ Pasos de ejecución: Se intenta modificar una tarea asignada por el mismo a otro profesor pasando los datos correctamente	
Resultado esperado: Se muestra la tarea con el cambio realizado en el listado de tareas creadas por el mismo.	
Evaluación de la prueba: Satisfactoria	

Tabla 77 Prueba 6 de la HU “Administrar tareas”.

Caso de prueba de aceptación:	
Código: HU3_p6	Historia de Usuario: 3
Nombre: Eliminar una tarea asignada	
Descripción: Probar la funcionalidad eliminar una tarea asignada por el usuario.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de desempeño.	
Entrada/ Pasos de ejecución: Se intenta eliminar una tarea asignada por el mismo a otro profesor pasando los datos correctamente	
Resultado esperado: Se muestra el listado de tareas asignadas por el usuario sin la tarea que se eliminó.	
Evaluación de la prueba: Satisfactoria	

- HU “Administrar desempeño”

Tabla 78 Prueba 1 de la HU “Administrar desempeño”.

Caso de prueba de aceptación:	
Código: HU4_p1	Historia de Usuario: 4

Nombre: Añadir ausencias
Descripción: Probar la funcionalidad añadir ausencias a determinado(s) profesor(es).
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de desempeño.
Entrada/ Pasos de ejecución: Se intenta añadir una ausencia de una actividad insertando los datos correctamente (sin dejar algún campo requerido en blanco).
Resultado esperado: Se le muestra al usuario la ausencia añadida en el listado de las ausencias que el mismo ha reportado a sus profesores.
Evaluación de la prueba: Satisfactoria

Tabla 79 Prueba 2 de la HU "Administrar desempeño".

Caso de prueba de aceptación:	
Código: HU4_p2	Historia de Usuario: 4
Nombre: Añadir incidencia	
Descripción: Probar la funcionalidad añadir incidencia a determinado(s) profesor(es).	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de desempeño.	
Entrada/ Pasos de ejecución: Se intenta reportar una incidencia insertando para ellos los datos pertinentes (sin dejar algún campo requerido en blanco).	
Resultado esperado: Se le muestra al usuario la incidencia en el listado de las incidencias que el mismo ha reportado a sus profesores.	
Evaluación de la prueba: Satisfactoria	

Tabla 80 Prueba 3 de la HU "Administrar desempeño".

Caso de prueba de aceptación:	
Código: HU4_p3	Historia de Usuario: 4
Nombre: Evaluar tarea	
Descripción: Probar la funcionalidad evaluar tarea.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de tarea.	

Entrada/ Pasos de ejecución: Se intenta evaluar una tarea correctamente.
Resultado esperado: Se le muestra al usuario el listado de las tareas a evaluar sin la que se ha evaluado, poniendo a la misma con su respectiva evaluación en el listado de tareas evaluadas del profesor al que se le ha evaluado.
Evaluación de la prueba: Satisfactoria

Tabla 81 Prueba 4 de la HU “Administrar desempeño”.

Caso de prueba de aceptación:	
Código: HU4_p4	Historia de Usuario: 4
Nombre: Evaluar área	
Descripción: Probar la funcionalidad evaluar desempeño de un profesor en un área.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de área.	
Entrada/ Pasos de ejecución: Se intenta evaluar un área correctamente.	
Resultado esperado: Se añade el profesor al listado de los evaluados en esa área.	
Evaluación de la prueba: Satisfactoria	

Tabla 82 Prueba 5 de la HU “Administrar desempeño”.

Caso de prueba de aceptación:	
Código: HU4_p5	Historia de Usuario: 4
Nombre: Evaluar desempeño	
Descripción: Probar la funcionalidad evaluar desempeño de un profesor.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del evaluador de desempeño.	
Entrada/ Pasos de ejecución: Se intenta evaluar el desempeño de un profesor, seleccionando para ello la evaluación correspondiente.	
Resultado esperado: Se muestra el desempeño con su correspondiente evaluación	
Evaluación de la prueba: Satisfactoria	

- HU “Aprobar evaluación”

Tabla 83 Prueba 1 de la HU “Aprobar evaluación”.

Caso de prueba de aceptación:	
Código: HU5_p1	Historia de Usuario: 5
Nombre: Aprobar evaluación	
Descripción: Probar la funcionalidad Aprobar evaluación de un profesor.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del aprobador de evaluación.	
Entrada/ Pasos de ejecución: Se intenta aprobar la evaluación de desempeño de un profesor.	
Resultado esperado: Se le muestra al usuario la evaluación aprobada.	
Evaluación de la prueba: Satisfactoria	

- HU “Administrar usuario”

Tabla 84 Prueba 1 de la HU “Administrar usuario”.

Caso de prueba de aceptación:	
Código: HU6_p1	Historia de Usuario: 6
Nombre: Insertar profesor	
Descripción: Probar la funcionalidad insertar un nuevo profesor en la aplicación.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar un nuevo profesor con todos sus datos correctamente válidos.	
Resultado esperado: Se muestra el nuevo profesor añadido en el listado de profesores del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 85 Prueba 2 de la HU “Administrar usuario”.

Caso de prueba de aceptación:	
Código: HU6_p2	Historia de Usuario: 6
Nombre: Modificar profesor	
Descripción: Probar la funcionalidad modificar un profesor existente.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar profesor seleccionado	
Resultado esperado: Se muestra el profesor en el listado de profesores del sistema con los nuevos cambios.	
Evaluación de la prueba: Satisfactoria	

Tabla 86 Prueba 3 de la HU “Administrar usuario”.

Caso de prueba de aceptación:	
Código: HU6_p3	Historia de Usuario: 6
Nombre: Eliminar profesor	
Descripción: Probar la funcionalidad eliminar un profesor existente.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar profesor seleccionado	
Resultado esperado: Se muestra el listado de profesores del sistema sin el profesor que se eliminó.	
Evaluación de la prueba: Satisfactoria	

Tabla 87 Prueba 4 de la HU “Administrar usuario”.

Caso de prueba de aceptación:	
Código: HU6_p4	Historia de Usuario: 6
Nombre: Buscar profesor	
Descripción: Probar la funcionalidad buscar profesor.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los	

privilegios del administrador.
Entrada/ Pasos de ejecución: Se intenta buscar profesor pasando los parámetros de un profesor existente
Resultado esperado: Se muestra el profesor.
Evaluación de la prueba: Satisfactoria

Tabla 88 Prueba 5 de la HU "Administrar usuario".

Caso de prueba de aceptación:	
Código: HU6_p5	Historia de Usuario: 6
Nombre: Insertar cargo	
Descripción: Probar la funcionalidad insertar un nuevo cargo.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar un cargo nuevo pasando los datos a la aplicación correctamente.	
Resultado esperado: Se muestra el nuevo cargo en el listado de cargos del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 89 Prueba 6 de la HU "Administrar usuario".

Caso de prueba de aceptación:	
Código: HU6_p6	Historia de Usuario: 6
Nombre: Modificar un cargo	
Descripción: Probar la funcionalidad modificar un cargo existente	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar el cargo seleccionado.	
Resultado esperado: Se muestra el cargo con las modificaciones hechas en el listado de cargos del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 90 Prueba 7 de la HU "Administrar usuario".

Caso de prueba de aceptación:	
Código: HU6_p7	Historia de Usuario: 6
Nombre: Eliminar un cargo	
Descripción: Probar la funcionalidad eliminar un cargo.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar un cargo	
Resultado esperado: Se muestra el listado de cargos del sistema sin el cargo eliminado.	
Evaluación de la prueba: Satisfactoria	

Tabla 91 Prueba 8 de la HU "Administrar usuario".

Caso de prueba de aceptación:	
Código: HU6_p8	Historia de Usuario: 6
Nombre: Insertar departamento	
Descripción: Probar la funcionalidad insertar departamento.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar un nuevo departamento pasando los datos correctamente.	
Resultado esperado: Se muestra el nuevo departamento en el listado de departamentos del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 92 Prueba 9 de la HU "Administrar usuario".

Caso de prueba de aceptación:	
Código: HU6_p9	Historia de Usuario: 6
Nombre: Modificar un departamento	
Descripción: Probar la funcionalidad modificar un departamento existente.	

Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.
Entrada/ Pasos de ejecución: Se intenta modificar un departamento
Resultado esperado: Se muestra el departamento con sus nuevos cambios en el listado de departamentos del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 93 Prueba 10 de la HU “Administrar usuario”.

Caso de prueba de aceptación:	
Código: HU6_p10	Historia de Usuario: 6
Nombre: Eliminar un departamento	
Descripción: Probar la funcionalidad eliminar un departamento.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar un departamento	
Resultado esperado: Se muestra el listado de departamentos del sistema sin el departamento eliminado.	
Evaluación de la prueba: Satisfactoria	

- HU “Administrar tipos de tareas”

Tabla 94 Prueba 1 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p1	Historia de Usuario: 7
Nombre: Insertar área	
Descripción: Probar la funcionalidad insertar área nueva.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar un área con los datos válidos	
Resultado esperado: Se muestra la nueva área en el listado de áreas del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 95 Prueba 2 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p2	Historia de Usuario: 7
Nombre: Modificar un área	
Descripción: Probar la funcionalidad modificar un área.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar un área	
Resultado esperado: Se muestra el área modificada en el listado de áreas del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 96 Prueba 3 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p3	Historia de Usuario: 7
Nombre: Eliminar un área	
Descripción: Probar la funcionalidad eliminar un área.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar un área	
Resultado esperado: Se muestra el listado de áreas del sistema sin el área eliminada.	
Evaluación de la prueba: Satisfactoria	

Tabla 97 Prueba 4 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p4	Historia de Usuario: 7
Nombre: Insertar tipo de tarea	
Descripción: Probar la funcionalidad insertar tipo de tarea.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los	

privilegios del administrador.
Entrada/ Pasos de ejecución: Se intenta insertar un tipo de tarea con los datos correctos
Resultado esperado: Se muestra un nuevo tipo de tarea en el listado de tipos de tarea del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 98 Prueba 5 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p5	Historia de Usuario: 7
Nombre: Modificar un tipo de tarea	
Descripción: Probar la funcionalidad modificar un tipo de tarea	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar un tipo de tarea	
Resultado esperado: Se muestra el tipo de tarea modificado en el listado de tipos de tarea del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 99 Prueba 6 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p6	Historia de Usuario: 7
Nombre: Eliminar un tipo de tarea	
Descripción: Probar la funcionalidad eliminar un tipo de tarea.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar un tipo de tarea	
Resultado esperado: Se muestra el listado de tipos de tarea del sistema sin el tipo de tarea eliminado.	
Evaluación de la prueba: Satisfactoria	

Tabla 100 Prueba 7 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p7	Historia de Usuario: 7
Nombre: Insertar actividad	
Descripción: Probar la funcionalidad insertar actividad.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar una actividad con los datos válidos	
Resultado esperado: Se muestra la nueva actividad en el listado de actividades del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 101 Prueba 8 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p8	Historia de Usuario: 7
Nombre: Modificar una actividad	
Descripción: Probar la funcionalidad modificar una actividad.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar una actividad	
Resultado esperado: Se muestra la actividad modificada en el listado de actividades del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 102 Prueba 9 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p9	Historia de Usuario: 7
Nombre: Eliminar una actividad	
Descripción: Probar la funcionalidad eliminar una actividad	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los	

privilegios del administrador.
Entrada/ Pasos de ejecución: Se intenta eliminar una actividad
Resultado esperado: Se muestra el listado de actividades del sistema sin la actividad eliminada.
Evaluación de la prueba: Satisfactoria

Tabla 103 Prueba 10 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p10	Historia de Usuario: 7
Nombre: Insertar tarea para el plan de resultados	
Descripción: Probar la funcionalidad insertar tarea para el plan de resultados.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar una tarea con los datos válidos	
Resultado esperado: Se muestra la nueva tarea en el listado de tareas del plan de resultados.	
Evaluación de la prueba: Satisfactoria	

Tabla 104 Prueba 11 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p11	Historia de Usuario: 7
Nombre: Modificar una tarea del plan de resultados	
Descripción: Probar la funcionalidad modificar una tarea del plan de resultados.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar una tarea del plan de resultados.	
Resultado esperado: Se muestra la tarea modificada en el listado de tareas del plan de resultados.	
Evaluación de la prueba: Satisfactoria	

Tabla 105 Prueba 12 de la HU “Administrar tipos de tareas”.

Caso de prueba de aceptación:	
Código: HU7_p12	Historia de Usuario: 7
Nombre: Eliminar una tarea del plan de resultados	
Descripción: Probar la funcionalidad eliminar una tarea del plan de resultados	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar una tarea del plan de resultados	
Resultado esperado: Se muestra el listado de tareas del plan de resultados sin dicha tarea eliminada.	
Evaluación de la prueba: Satisfactoria	

- HU “Administrar sistema”

Tabla 106 Prueba 1 de la HU “Administrar sistema”.

Caso de prueba de aceptación:	
Código: HU8_p1	Historia de Usuario: 8
Nombre: Insertar rol	
Descripción: Probar la funcionalidad insertar rol	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar un rol con datos correctos	
Resultado esperado: Se muestra el nuevo rol en el listado de roles del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 107 Prueba 2 de la HU “Administrar sistema”.

Caso de prueba de aceptación:	
Código: HU8_p2	Historia de Usuario: 8
Nombre: Modificar un rol	
Descripción: Probar la funcionalidad modificar un rol	

Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.
Entrada/ Pasos de ejecución: Se intenta modificar el rol seleccionado
Resultado esperado: Se muestra el rol modificado en el listado de roles del sistema.
Evaluación de la prueba: Satisfactoria

Tabla 108 Prueba 3 de la HU "Administrar sistema".

Caso de prueba de aceptación:	
Código: HU8_p3	Historia de Usuario: 8
Nombre: Eliminar un rol	
Descripción: Probar la funcionalidad eliminar un rol	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar un rol	
Resultado esperado: Se muestra el listado de roles del sistema sin el rol eliminado.	
Evaluación de la prueba: Satisfactoria	

Tabla 109 Prueba 3 de la HU "Administrar sistema".

Caso de prueba de aceptación:	
Código: HU8_p1	Historia de Usuario: 8
Nombre: Insertar página principal	
Descripción: Probar la funcionalidad insertar página principal	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta insertar una página principal correctamente	
Resultado esperado: Se muestra la nueva página principal en el listado de páginas principal del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 110 Prueba 2 de la HU "Administrar sistema".

Caso de prueba de aceptación:	
Código: HU8_p2	Historia de Usuario: 8
Nombre: Probar la funcionalidad modificar página principal	
Descripción: Probar la funcionalidad modificar página principal.	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta modificar una página principal	
Resultado esperado: Se muestra la página principal modificada en el listado de páginas principal del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 111 Prueba 3 de la HU "Administrar sistema".

Caso de prueba de aceptación:	
Código: HU8_p3	Historia de Usuario: 8
Nombre: Eliminar una página principal	
Descripción: Probar la funcionalidad eliminar una página principal	
Condiciones de ejecución: El usuario debe estar autenticado y debe tener los privilegios del administrador.	
Entrada/ Pasos de ejecución: Se intenta eliminar una página principal	
Resultado esperado: Se muestra el listado de páginas principal del sistema sin la página principal eliminada.	
Evaluación de la prueba: Satisfactoria	

- HU "Generar Reporte"

Tabla 112 Prueba 1 de la HU "Generar Reporte".

Caso de prueba de aceptación:	
Código: HU9_p1	Historia de Usuario: 9
Nombre: Generar Reporte	

Descripción: Probar la funcionalidad generar un reporte del desempeño
Condiciones de ejecución: El usuario debe estar autenticado
Entrada/ Pasos de ejecución: Se intenta generar un reporte del desempeño indicándole el trimestre.
Resultado esperado: Se genera un documento con el reporte del desempeño de ese profesor en ese trimestre especificado.
Evaluación de la prueba: Satisfactoria

- HU “Crear plan de resultados”

Tabla 113 Prueba 1 de la HU “Crear plan de resultados”.

Caso de prueba de aceptación:	
Código: HU10_p1	Historia de Usuario: 10
Nombre: Crear plan de resultados	
Descripción: Probar la funcionalidad crear el plan de resultados de un profesor.	
Condiciones de ejecución: El usuario debe estar autenticado.	
Entrada/ Pasos de ejecución: Se intenta crear su plan de resultados.	
Resultado esperado: Las tareas son creadas y guardadas con éxito.	
Evaluación de la prueba: Satisfactoria	

En resumen, en este capítulo se hizo referencia a todo lo concerniente a la fase de Prueba de la metodología XP. Se realizó la descripción de las pruebas de aceptación a la que fue sometida la aplicación con la intención de verificar el funcionamiento de la misma, probando individualmente la implementación de cada historia de usuario en su correspondiente iteración y asignándole una evaluación.

Todas las pruebas realizadas fueron aceptadas por el cliente cumpliendo entonces el sistema con las historias de usuarios definidas inicialmente.

Conclusiones

Para el cumplimiento de los objetivos y en concordancia con las necesidades expuestas por el cliente, se requirió hacer un estudio de la situación actual del proceso de gestión de evaluación de desempeño de los profesores de la facultad 10 de la Universidad de las Ciencias Informáticas, para con ello elevar la eficiencia del proceso en cuestión

Mediante la realización de un estudio sobre la ingeniería de software más acertada para ser utilizada se determinó utilizar XP y como lenguaje de modelado UML, esta metodología ayudó a guiar el proceso de diseño de la propuesta del sistema, además de permitir que el trabajo se desarrollara con mayor eficacia.

Se realizó un estudio del reglamento sobre la aplicación del sistema para pago adicional.

Se caracterizaron las herramientas y lenguajes de programación adecuados para el desarrollo de la aplicación.

Se implementó un proceso modelo para la gestión de las evaluaciones de desempeño de los profesores de la facultad 10.

Se desarrolló un sistema automatizado capaz de gestionar con mayor eficiencia el proceso de evaluación de desempeño.

Recomendaciones

Luego de haber implementado la propuesta para el sistema automatizado que gestiona la evaluación de desempeño de los profesores se recomienda:

- ✓ Utilizar este trabajo de diploma como bibliografía para posteriores investigaciones.
- ✓ Se incluya el software implementado en el proceso de gestión de evaluación de desempeño llevado a cabo en la facultad 10 de la Universidad de las Ciencias Informáticas.
- ✓ Implementar otros módulos que contribuyan a facilitar aún más el trabajo en el proceso.
- ✓ Continuar realizando mejoras a la arquitectura de la información, así como al diseño.

Referencia bibliográfica

1. **ABC., Definición.** Definición de Gestión. [En línea] 2009. [Citado el: 24 de Enero de 2010.] <http://www.definicionabc.com/general/gestion.php>.
2. **Farlex, Inc.** Proceso. [En línea] 2010. [Citado el: 15 de Febrero de 2010.] <http://es.thefreedictionary.com/proceso>.
3. **ABC., Definición.** Definición de eficiencia. [En línea] [Citado el: 01 de Marzo de 2010.] <http://definicion.de/eficiencia/>.
4. **ABC., Definición.** Definición de evaluación. [En línea] 2009. [Citado el: 02 de Febrero de 2010.] <http://www.definicionabc.com/general/evaluacion.php>.
5. **Cagua., Fernando.** Definición de automatización. [En línea] [Citado el: 16 de Diciembre de 2009.] <http://www.mecatronica-portal.com/2009/04/129-definicion-de-automatizacion/>.
6. **Lanzillotta., Analía.** Definición de Información. [En línea] 2007. [Citado el: 14 de Diciembre de 2009.] <http://www.mastermagazine.info/termino/5366.php>.
7. **Aja Quiroga, Lourdes.** Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. [En línea] 2002. [Citado el: 21 de Marzo de 2010.] http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm.
8. **Gutiérrez, Jorge A. Saavedra.** El Mundo Informático. Software Libre. [En línea] 2007. [Citado el: 05 de Marzo de 2010.] <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>.
9. **Murugarren, Joaquin Gracia.** WebEstilo. [En línea] 2008. [Citado el: 22 de Enero de 2010.] <http://www.webestilo.com>.
10. **Quesada, Juan de.** Universidad de las Palmas Gran Canaria. Universidad de las Palmas Gran Canaria. [En línea] 2006. [Citado el: 16 de Diciembre de 2009.] <http://www.ulpgc.es/otros/tutoriales..>
11. **Web., World Wide.** World Wide Web. [En línea] 2008. [Citado el: 23 de Febrero de 2010.] <http://www.w3c.es>.
12. **González, Rubén Marcos.** Recursos para la programación en php. [En línea] 2010. [Citado el: 02 de Diciembre de 2009.] http://tgp0607.awardspace.com/Recursos_PHP.pdf.
13. **Piñol, Carles Mateu.** Cibernética. [En línea] 2010. [Citado el: 17 de Diciembre de 2010.] <http://www.cibernetia.com/manuales/>.
14. **Ciberaula.** Una introducción a Apache. [En línea] 2006. [Citado el: 18 de Enero de 2010.] http://linux.ciberaula.com/articulo/linux_apache_intro/.

15. **Web.net., Programación.** Introducción a MySQL. [En línea] 2006. [Citado el: 24 de Febrero de 2010.] <http://www.programacionweb.net/articulos/articulo/?num=184>.
16. **Martínez, Rafael.** Sobre PostgreSQL. [En línea] 2006. [Citado el: 12 de Diciembre de 2009.] http://www.postgresql-es.org/sobre_postgresql.
17. **Díaz, Néstor.** Ventajas de PostgreSQL. [En línea] 2010. [Citado el: 24 de Febrero de 2010.] http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas_html.
18. **Potencier, Fabien.** Symfony 1.2, la guía definitiva. [En línea] 2009. [Citado el: 26 de Marzo de 2010.] http://www.librosweb.es/symfony_1_2/.
19. **Molpeceres, Alberto.** Proceso de Desarrollo. [En línea] 2009. [Citado el: 19 de Febrero de 2010.] http://www.javahispano.org/contenidos/es/procesos_de_desarrollo/.
20. **Gálvez, Jorge.** Fundamentos de la metodología RUP. [En línea] 2009. [Citado el: 22 de Enero de 2010.] <http://www.scribd.com/doc/13071186/Rup>.
21. **James Rumbaugh, Ivar Jacobson, Grady Booch.** El Proceso Unificado de Desarrollo de Software. [En línea] [Citado el: 16 de Enero de 2010.] <http://bibliodoc.uci.cu/pdf/reg00060.pdf>.
22. **olpeceres, Alberto.** M Procesos de desarrollo: RUP, XP y FDD.,. [En línea] 2003. [Citado el: 16 de Diciembre de 2009.] <http://www.javahispano.org/articles.article.action?id=76>.
23. **Hernández, Orallo.** El Lenguaje Unificado de Modelado (UML). [En línea] 2007. [Citado el: 07 de Febrero de 2010.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
24. **Larman, C. UML y Patrones.** Introducción al análisis y diseño orientado a objetos. [En línea] 2007. [Citado el: 21 de Diciembre de 2009.] <http://bibliodoc.uci.cu/pdf/reg00061.pdf>.
25. **Álvarez, Miguel Ángel.** Zend Studio. [En línea] 2003. [Citado el: 12 de Febrero de 2010.] <http://www.desarrolloweb.com/articulos/1178.php>.
26. **Canales, Mora.** Adictos al Trabajo. [En línea] 2004. [Citado el: 13 de Marzo de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=vparadigm..>
27. **León, Eduardo.** Tutorial Visual Paradigm. [En línea] 2008. [Citado el: 15 de Diciembre de 2009.] <http://sliion2000.blogspot.com/2008/10/tutorial-visual-paradigm.html>.

Bibliografía

- ✓ **Aguilar, A.** Programación Extrema y Software Libre. [En línea] 2002. [Citado el: 03 de Abril de 2010.]
<http://www.seguridad.unam.mx/eventos/datos/ev11/semi18/mat.7.pon19.semi18.pdf>.
- ✓ **Calero Solís, Manuel. Apolo Software.** Una explicación de Programación extrema. [En línea] [Citado el: 14 de Enero de 2010.] <http://www.apolosoftware.com/>.
- ✓ **Carlos., Reynoso Billy.** Metodos Agiles en Desarrollo de Software, Introducción a la Arquitectura de Software. Universidad de Buenos Aires. [En línea] [Citado el: 12 de Enero de 2010.] <http://www.microsoft.com/spanish/msdn/arquitectura..>
- ✓ **E.Hernández.** El Lenguaje Unificado de Modelado (UML),. [En línea] 2004. [Citado el: 24 de Febrero de 2010.] http://www.acta.es/articulos_mf/26067.pdf.
- ✓ **Figueroa, R., Solís, C. y Cabrera, A.** Metodologías Tradicionales Vs. Metodologías Ágiles. [En línea] 2008. [Citado el: 26 de Febrero de 2010.]
<http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato..>
- ✓ **José, H. C. Patricio, L. María Carmen, P.** Metodologías Ágiles en el Desarrollo de Software. [En línea] 2007. [Citado el: 21 de Febrero de 2010.] Disponible en:
www.willydev.net/descargas/prev/TodoAgil.Pdf.
- ✓ **José H. Canós, P. L., Mª Carmen Penadés.** (XP) Metodologías Ágiles en el Desarrollo de Software. [En línea] 2002. [Citado el: 04 de Abril de 2010.] from
<http://www.willydev.net/descargas/prev/TodoAgil.Pdf..>
- ✓ **León, Rolando Alfredo Hernández.** EL PARADIGMA CUANTITATIVO. [En línea] 2002. [Citado el: 18 de Diciembre de 2009.]
<http://revistas.mes.edu.cu/elibro/libros/300/9789591603432.pdf/view>.
- ✓ **Meléndrez, Edelsys Hernández.** Como escribir una tesis. [En línea] 2006. [Citado el: 12 de Febrero de 2010.]
[http://eva.uci.cu/file.php/63/Otros_Materiales_de_Consulta/Como_escribir_una_tesis.p
df](http://eva.uci.cu/file.php/63/Otros_Materiales_de_Consulta/Como_escribir_una_tesis.pdf).
- ✓ **M.Valverde, D.** Introducción a la programación extrema rápidos. [En línea] 2007. [Citado el: 03 de Abril de 2010.] www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp.
- ✓ **O., Hagstrom W.** El don como principio organizado de la ciencia. Madrid, España. : Alianza Universidad, 1980.

- ✓ **R., Torrens.** Arquitectura de la información. [En línea] 1999. [Citado el: 24 de Febrero de 2010.]
<http://oai.saber.ula.ve/db/ssaber/Edocs/postgrados/computacion/Arquitecturadelainformacion.pdf>.
- ✓ **Solís., Manuel Calero.** Una explicación de la programación extrema (XP). V *Encuentro usuarios xBase*. [En línea] 2003. [Citado el: 19 de Diciembre de 2009.]
<http://www.apolosoftware.com/>.
- ✓ **Zayas, Dr. Cs. Carlos Alvarez de.** METODOLOGIA DE LA INVESTIGACION CIENTIFICA. [En línea] 1995. [Citado el: 26 de Diciembre de 2009.]
http://eva.uci.cu/file.php/63/Otros_Materiales_de_Consulta/Carlos_Alvarez_de_Zayas_-_Libro.pdf.